UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA

Recolha de cyberthreat open souce intelligence usando técnicas de correlação de informação

Edgar Tito Matias Oliveira Martins

Mestrado em Segurança Informática

Dissertação orientada por:
Professor Doutor António Casimiro

2019

Agradecimentos

A realização desta dissertação de mestrado contou com importantes apoios e incentivos sem os quais muito mais difícil seria de a alcançar e isso é motivo de grande gratidão.

Ao Professor Doutor António Casimiro, pela sua orientação, total apoio e disponibilidade, pelo saber e opinião crítica na resolução de todos os problemas que foram surgindo ao longo da realização deste trabalho, nunca faltando uma palavra de incentivo.

Uma palavra especial aos colegas da equipa do COSI, especialmente o Eng. Jean Figueiredo e o Eng. Carlos Torres pela disponibilidade e colaboração manifestadas.

Quero agradecer ainda a todos os meus colegas de faculdade e amigos que estiveram sempre do meu lado nos momentos bons e em alguns momentos mais difíceis, o seu companheirismo merece o meu reconhecimento.

Por fim e não menos importante quero fazer um agradecimento especial aos meus pais e à minha irmã pelo apoio incondicional no meu percurso académico e pessoal. Quero fazer um sublinhado especial ao apoio da minha mulher e à minha mãe pela especial importância que tiveram para que este marco da minha vida académica se concretizasse.

Resumo

Face ao rápido desenvolvimento do ciberespaço, bem como de todas as infra-estruturas e sistemas digitais a ele associados, tornou-se clara a necessidade de preparar as organizações para esta evolução. A dependência das tecnologias de informação e o aumento do número de sistemas interligados entre si vieram trazer novas oportunidades para os que pretendem comprometer os sistemas digitais que existem na actualidade.

O número de ciberataques, bem como a sua complexidade, tem vindo a aumentar e está a tornar-se uma grande ameaça para a ambiente digital das organizações, que se tentam proteger destas ameaças através do investimento num Centro de Operações de Segurança (SOC).

No contexto de um SOC é conhecida a importância de ter os sistemas actualizados com a informação mais recente disponível online. A informação Open Source Intelligence (OSINT) é cada vez mais valiosa, devido à importância que tem na prevenção de ataques num sistema de monitorização em tempo real.

A quantidade de indicadores de compromisso recolhida pelas aplicações de recolha de informação OSINT tem vindo a aumentar pelo que se torna difícil avaliar a eficácia dos dados recolhidos por estas plataformas. A necessidade de estar actualizado com a informação mais recente sobre os ataques informáticos é cada vez maior e tornou-se num problema em SOCs que monitorizam ambientes informáticos de grande escala.

Por isso criámos a Zwerg, um software que recolhe indicadores de compromisso a partir de notícias e agrega metadados a esses indicadores. O objectivo do trabalho descrito neste projecto é recolher indicadores de compromisso e enriquecê-los com metainformação obtida a partir de dados recolhidos de aplicações web disponíveis ao público. A informação recolhida pela Zwerg é extremamente valiosa para detecção e prevenção de ataques informáticos.

Este projecto foi desenvolvido e testado no Centro de Operações de Segurança Informática (COSI) do Ministério da Administração Interna (MAI) e os resultados obtidos revelaram o grande valor acrescentado e importância que o software Zwerg tem para o SOC do MAI.

Palavras chave: OSINT, indicadores de compromisso, MISP, SOC, cibermeaças.

Abstract

In view of the rapid development of cyberspace, as well as of all associated digital infrastructures and systems, the need to prepare organizations for this evolution has become clear. The current dependency on information technologies and the fast pace increase in the number of interconnected systems have provided new opportunities for those who wish to compromise the current digital systems.

The number of cyberattacks, and their complexity, is increasing and is becoming a major threat to the digital environment of organizations that are attempting to protect themselves against such threats by investing in a Security Operations Center (SOC).

In the context of a SOC it is known the importance of having the systems updated with the latest information available online. Open Source Intelligence (OSINT) information is increasingly valuable because of the importance it plays in preventing attacks on a real-time monitoring system.

The number of indicators of compromise collected by OSINT information collection applications has been increasing, making it difficult to assess the effectiveness of the data collected by these platforms. The need to be updated with the latest information on computer attacks is increasing and has become a problem in SOCs that monitor large-scale computing environments.

That's why we created Zwerg, a software that collects indicators of compromise from news and aggregates metadata to these indicators. The objective of the work described in this dissertation is to collect collects indicators of compromise and enrich them with metadata obtained from data collected from web applications available to the general public. The information collected by Zwerg is extremely valuable for detecting and preventing computer attacks.

This project was developed and tested at the Centro de Operações de Segurança Informática (COSI) of Ministério da Administração Interna (MAI) and the results revealed the great added value and importance that the Zwerg software has for the SOC of MAI.

Keywords: OSINT, indicators of compromise, MISP, SOC, cyber threats.

Conteúdo

Lista de Figuras

Lista de Tabelas

Lista de Abreviações

API	*Application Programming Interface*
GB	*Gigabytes*
CPU	*Central Processing Unit*
COSI	*Centro de Operações de Segurança Informática*
CSIRT	*Computer Security Incident Response Team*
CSS	*Cascade Style Sheets*
DDoS	*Distributed Denial of Service*
DNS	*Domain Name System*
DoS	*Denial of Service*
ENISA	*European Union Agency for Cybersecurity*
HTML	*Hypertext Markup Language*
HTTP	*Hypertext Transfer Protocol*
IOC	*Indicator of Compromise*
IP	*Internet Protocol*
IDS	*Intrusion Detections Systems*
IPS	*Intrusion Prevention Systems*
ISP	*Internet Service Provider*
JSON	*JavaScript Object Notation*
MAI	*Ministério da Administração Interna*
MISP	*Malware Information Sharing Platform*
MongoDB	*Mongo Database*
NMAP	Network Mapper
OSINT	*Open Source Intelligence*
PCRE	*Perl Compatible Regular Expressions*
PDF	*Portable Document Format*
RNSI	*Rede Nacional de Segurança Interna*
RSS	*Really Simple Syndication*
SIEM	*Security Information and Event Management*
SO	*Sistemas Operativos*
SOC	*Security Operation Center*
SQL	*Structured Query Language*

URI	*Unique Resource Identfication*
URL	*Unique Resource Location*
XML	*Extended Markup Language*
WAF	*Web Application Firewall*

Capítulo 1 **Introdução**

A evolução dos sistemas digitais, a um ritmo acelerado, veio transformar a forma como interagimos com o mundo digital, em especial com o ciberespaço. O aumento da capacidade computacional das máquinas fez com que os humanos confiassem cada vez mais o processamento de tarefas às máquinas [1]. Como consequência houve uma massificação do uso deste tipo de equipamentos, também conhecida como era digital. O acesso generalizado à Internet veio trazer enormes benefícios para todos os utilizadores, pois não só se tornou mais fácil a comunicação e a troca de dados, como também se geraram novos sectores de negócio. Assim, todas as grandes empresas começaram a disponibilizar pequenos serviços na Internet por forma a alcançar os consumidores que representavam o seu público alvo.

Nos dias de hoje existe uma grande quantidade de informação que se pode encontrar na Internet. Por dia são gerados aproximadamente 5 biliões de *gigabytes* (GB) de tráfego na Internet [2]. Os sistemas informáticos, onde está inserida toda esta informação, são construídos com base em *frameworks* e linguagens de programação em constante evolução. A complexidade destes sistemas faz com que, por vezes, sejam descobertas falhas que podem ter consequências nefastas, como por exemplo o comprometimento dos mecanismos de controlo de acesso a informação, a execução de procedimentos programáticos inapropriados que podem causar dano ou indisponibilidade nos sistemas do ambiente alvo, entre outras. A descoberta deste tipo de vulnerabilidades tem proporcionado o aumento do número de ciberataques bem como o seu grau de exploração. Os ataques provêm de diversos tipos de agentes, desde o atacante que actua sozinho até aos grupos de crime organizado.

Com a massificação do uso da Internet, em sociedades conectadas, surgem cada vez mais utilizadores, com pouca experiência na administração e gestão de sistemas informáticos complexos onde são armazenadas grandes quantidades de informação. Estas condições tornam estes sistemas mais vulneráveis a ataque informáticos perpetrados por indivíduos ou organizações criminosas, conhecidos como hackers. A palavra inglesa *hack* significa a arte de encontrar uma solução astuta e elegante para um problema incomum ou não documentado [3].

Apesar dos constantes avanços tecnológicos não existem actualmente sistemas completamente seguros. Tendo em conta a abundância e a diversidade e complexidade das vulnerabilidades existentes nos sistemas informáticos, as organizações não se conseguem manter imunes a ataques. Para além disso, diariamente são descobertas novas vulnerabilidades nos Sistemas Operativos (SOs) e nas aplicações, proprietárias ou de software aberto, com variados níveis de criticidade e potencial impacto quando exploradas.

No processo de evolução que, culminou no "mundo digital", a segurança dos sistemas, ambientes informáticos, organizações e utilizadores nem sempre foi levada em conta como pilar essencial sobre o qual assenta a construção dos sistemas informáticos. Como consequência, as vulnerabilidades ocorrem a vários níveis,

nomeadamente ao nível da arquitectura dos sistemas, da sua programação, da sua configuração , entre outros [4].

Actualmente, tendo em conta a complexidade dos sistemas, frequentemente distribuídos e interligados através de múltiplas redes, e verificando-se um aumento contínuo do número de ataques informáticos registados anualmente [5], a necessidade de mecanismos e soluções que garantam a segurança dos sistemas e da informação tem vindo a revelar-se cada vez mais premente.

Segundo os dados da ESET, entre 2016 e 2017 observou-se um aumento histórico em ataques de ransomware que tiveram um impacto financeiro significativo nas grandes empresas. Neste período destacaram-se três grandes ataques: Wannacry Ransomware, DiskCoder.C e Industroyer malware. Segundo esses dados, em 2016 houve um aumento de 36% nos ataques de ransomware face ao ano anterior, em grande parte devido ao Wannacry [6].

Os ataques tipo ransomware são cada vez mais utilizados pelos atacantes como forma de financiamento, através de meios de pagamento que permitem transacções em aparente anonimato [7]. Os ataques de ransomware são extremamente poderosos e difíceis de detectar, pelo que é necessária uma detecção rápida e minuciosa para evitar danos avultados em sistemas de grande escala ou com elevado número de utilizadores.

A evolução da tecnologia e dos mercados *online* veio aumentar o estado de alerta em relação a novos tipos de ataques e novos alvos. Recentemente os ataques informáticos ligados á mineração de moeda digital (crytpominning) aumentaram de forma exponencial (34000% durante 2017) e revelaram novas ameaças e implicações financeiras, por exemplo, aos *datacenters* e *workstations* das empresas que recorrem a *cloud* Central Processing Units (CPU) [8].

A constante evolução dos ataques informáticos e a expansão das estruturas organizadas de cibercrime para novas áreas, fez com que as empresas comecem a alterar a sua postura, adoptando uma atitude proactiva em oposição à atitude reactiva, no que toca à prevenção de intrusões nos seus ambientes informáticos. Face a estes problemas, e considerando os números estatísticos referidos acima, as organizações têm-se prevenido com a formação e modernização das suas equipas de SOC. Estas equipas têm como principal objectivo detectar e prevenir ataques às infra-estruturas informáticas de uma organização. Para tornar o processo de detecção mais eficiente no SOC são usadas múltiplas ferramentas, entre as quais está o sistema Security Information and Event Management (SIEM). Esta ferramenta permite correlacionar grandes quantidades de informação por forma a detectar padrões de tráfego comuns e conhecidos de certos tipos de ataque. Para tornar cada vez mais eficaz a detecção destes padrões de tráfego através do SIEM é necessária uma constante actualização do conhecimento das equipas de SOC e dos mecanismos de detecção desses padrões.

Tendo em conta todos os argumentos referidos acima compreendemos a importância da busca pela informação associada aos ataques, que deve ser

actualizada e concreta para poder ser utilizada nos sistemas de SIEM, tornando o trabalho da equipa de SOC mais eficaz.

Por causa do aumento da quantidade e complexidade de dados presente no mundo digital tornou-se mais árdua a tarefa de selecção e extracção de informação concreta e relevante para o uso em sistemas SIEMs, em concreto, a extracção de indicadores de compromisso, ou Indicators of Compromise (IOCs) em inglês, correctos e válidos. Tudo isto torna importante a constante actualização das bases de dados de IOCs existentes num SOC, pois são a base de trabalho dos sistemas de análise dos *logs* que despoletam alertas de incidentes de segurança informática.

A importância que a informação e metainformação têm na actualidade, sobre os indicadores de compromisso, é um facto consumado. O valor da informação tem vindo a crescer, não só na área da segurança informática, mas com especial foco as áreas onde a reacção em tempo real é crítica e de valor acrescentado. É por isso importante contar com ferramentas que possam ajudar na recolha e processamento de dados por forma a transformá-los em informação útil e objectiva, neste caso, na manutenção de um SOC. A noção de dados é hoje muito específica e concreta em cada caso, isto é, cada equipa de SOC deseja ter um conjunto de dados relativos ao seu próprio ambiente, que na grande maioria das vezes é complexo e único.

Neste projecto propomo-nos construir um software capaz de recolher informação, processá-la e torná-la disponível para as diferentes ferramentas existentes num SOC através de um processo completamente automatizado. A aplicação desenvolvida, Zwerg, tem como objectivo processar a informação que se encontra disponível em diversos *websites* da especialidade por forma a obter um conjunto de dados organizados que possam ser utilizados na prevenção de ataques nos sistemas informáticos do Ministério da Administração Interna (MAI).

"No âmbito da prevenção, importa salvaguardar o papel fundamental da partilha de informação na avaliação precoce da ameaça. A incerteza permanente, relativa às diferentes ameaças de natureza difusa, contornos indefinidos e em permanente mutação e evolução que impendem sobre a segurança do ciberespaço de interesse nacional exigem capacidade nacional para detectar e conhecer atempadamente os indicadores que possam estar associados a ameaças potenciais e em curso. Neste sentido, é fulcral desenvolver a capacidade de obter, de forma automatizada, sistematizada e coerente, conhecimento desses indicadores. O conhecimento homogéneo e criterioso de indicadores de ameaça permitirá assim a todo o ecossistema nacional da segurança do ciberespaço o conhecimento prévio adequado à produção de medidas de antecipação da ameaça e de segurança contra impactos não desejados."

Diário da República, 1ª série – N.º – 5 de Junho de 2019 – Resolução do Conselho de Ministros n.º 92/2019

Durante o processo de investigação deste projecto foi possível perceber o cenário da realidade actual, bem como a necessidade, acima descrita, de desenvolver a capacidade de obter, de forma automatizada, sistematizada e coerente, conhecimento dos indicadores de compromisso. Este projecto vem responder a esta

necessidade existente em Portugal, no âmbito de prevenção, no contexto do SOC do MAI.

1.1 Motivação

A constante evolução dos ataques informáticos e a expansão do mundo do cibercrime para novas áreas, fez com que as empresas alterassem a sua atitude reactiva para uma atitude proactiva no que toca à prevenção de intrusões nos seus ambientes informáticos, investindo cada vez mais na formação e na modernização das suas equipas de SOC. Estas equipas têm comol principal objectivo detectar e prevenir ataques às infra-estruturas informáticas de uma organização, usando múltiplas ferramentas, entre as quais SIEMs. Estas ferramentas permitem correlacionar grandes quantidades de informação por forma a detectar padrões de tráfego característicos de certos tipos de ataque. Para tornar mais eficaz a detecção de ataques é necessária uma constante actualização do conhecimento das equipas de SOC relativamente aos padrões dos ataques. Para tal, é necessário recolher informação relativa a ataques de forma continuada, que deve ser actualizada e concreta para poder ser utilizada nos sistemas de SIEM, tornando o trabalho da equipa de SOC mais eficaz.

Por causa do aumento da quantidade e complexidade da informação que circula na Internet, a tarefa de selecção e recolha de informação concreta e relevante para o uso em sistemas SIEMs, nomeadamente a extracção de indicadores correctos e válidos, é uma tarefa difícil.

Actualmente existem diversas ferramentas que fornecem informação única sobre os diferentes tipos de ataques informáticos, bem como informação específica dos indícios disponíveis sobre cada um dos ataques. No entanto, a informação presente nessas plataformas depende muitas vezes de utilizadores que inserem informação com o objectivo de partilhar indicadores sobre os ataques informáticos. Este processo é maioritariamente um processo manual, na medida em que depende dos utilizadores e das listas de indicadores que eles criam e inserem nas plataformas. No entanto, existem diversos websites que disponibilizam IOCs em vários formatos, tais como listas ou artigos. Como não existem standards para a divulgação dos IOCs, nem sempre é possível obter através destas plataformas todos os indicadores disponíveis online sobre um determinado ataque ou activo.

Com base nestes argumentos surgiu a necessidade de construir um software que fosse capaz de corresponder às necessidades de recolha de informação objectiva para integração no ambiente de funcionamento do SOC da equipa do Centro de Operações de Segurança Informática (COSI) do Ministério da Administração Interna (MAI), no contexto do qual se realiza este trabalho.

Neste projecto propomo-nos construir uma ferramenta denominada Zwerg, capaz de recolher informação, processá-la e torná-la disponível para as outras ferramentas existentes num SOC de forma automatizada. O objectivo fundamental é utilizar a Zwerg na melhoria da prevenção de ataques aos sistemas informáticos do Ministério da Administração Interna (MAI).

1.2 Objectivos

O objectivo deste projecto é construir um software que automatiza o processo de recolha e tratamento de Open Source Intelligence (OSINT) para integração com o software existente no Centro de Operações de Segurança Informática da Rede Nacional de Segurança Interna (RNSI) do MAI. Este processo visa a recolha, interpretação, normalização e armazenamento de indicadores encontrados em websites, bem como a criação de metadados sobre esses indicadores. O software irá executar estes procedimentos de forma cíclica num período regular com um intervalo temporal definido pelo SOC. Os resultados obtidos serão armazenados numa BD local - Malware Information Sharing Platform (MISP).

Pretende-se, no fundo, que o software automatize o processo de recolha de IOCs, criação de respectivos metadados e, finalmente, armazenamento de toda a informação relevante numa instância MISP para integração com as Active Lists do SIEM. Desta forma será possível usar o SIEM para correlacionar os *logs* das máquinas existentes com os indicadores recolhidos. Com este software deverá ainda ser possível detectar as tendências actuais, no que diz respeito a ciberataques e vectores de ataque, através da detecção de padrões dos metadados associados aos indicadores recolhidos a partir das notícias publicadas pelos principais websites da especialidade.

1.3 Planeamento

O desenvolvimento deste projecto teve início a 15 de setembro de 2018, sendo que após essa data foi iniciado um trabalho contínuo e planeado para uma duração de nove meses.

Na primeira etapa do desenvolvimento deste projecto foi efectuado um estudo sobre o estado da arte para aferir os principais desafios a encontrar durante o desenvolvimento do projecto. Nesta fase foram estabelecidos os principais objectivos do projecto, a arquitectura da solução e o planeamento para o seu desenvolvimento e implementação. Esta etapa decorreu ao longo de um período de dois meses.

Numa segunda etapa foram explorados os vários conceitos necessários ao desenvolvimento da implementação técnica. Esta componente de desenvolvimento prático do projecto está organizada em diferentes secções: 1) Recolha de notícias a partir de uma lista de *feeds*; 2) Normalização e armazenamento dos dados numa Mongo Database (MongoDB); 3) Extracção de indicadores e criação de metadados sobre eles; 4) Processamento e normalização da informação extraída; 5) Criação de uma página Web para visualização de informação; 6) Integração dos indicadores e metadados no MISP. Esta etapa decorreu ao longo de um período de seis meses.

Na última fase deste projecto foram realizadas várias sessões de testes, durante o período de um mês, por forma avaliar a eficácia do software desenvolvido no contexto real do ambiente de SOC do MAI.

1.4 Contribuições

O presente trabalho oferece três contribuições: 1) O desenvolvimento de uma *framework* que suporta o software criado neste projecto; 2) A criação de uma aplicação que permite a total automatização de recolha de dados, análise de dados e criação de respectiva metainformação; 3) Desenvolvimento de uma ferramenta de visualização de dados disponível em permanência no *videowall* do COSI.

No desenvolvimento deste projecto foram criados vários procedimentos programáticos, bem como uma *framework* que suporta os diferentes conceitos aplicados no desenvolvimento da solução Zwerg. A *framework* utilizada neste projecto incide sobre os seguintes processos: recolha de conteúdo e processamento dos dados recolhidos através de software desenvolvido em Python; armazenamento de informação com recurso a uma base de dados MongoDB; visualização de informação a partir de uma aplicação web executada num browser, desenvolvida com recurso a JavaScript e Python; armazenamento dos indicadores recolhidos na plataforma MISP através da *Application Programming Interface* (API) desenvolvida em Python, "PyMISP".

Para o processo de automatização de recolha de dados foi importante escolher uma linguagem que disponibilizasse todas as ferramentas necessárias para uma implementação transversal a todas as aplicações usadas nos diferentes níveis. Outro factor essencial foi a utilização de uma linguagem de programação de alto nível para que o processo de elaboração da mesma se cingisse ao tempo útil disponível. Por estas razões a escolha recaiu sobre a linguagem de programação Python.

A escolha de MongoDB para armazenar os dados localmente deve-se ao facto de a base de dados MongoDB ser não relacional e permitir o armazenamento de grandes quantidades de dados de forma rápida e com menor percentagem de utilização de disco. Para além disso, disponibiliza uma API desenvolvida em Python para que possa ser realizada a integração com o MISP.

A aplicação web é desenvolvida com recurso a diversas linguagens de programação quer do lado do servidor, utilizando Python, quer do lado do cliente, utilizando HyperText Markup Language (HTML), Cascade Style Sheets (CSS) e Javascript.

1.5 Estrutura do documento

Este documento está divido em 7 capítulos. No **Capítulo 1**, na Introdução, é apresentada uma breve descrição dos tópicos inerentes ao tema abordado no projecto. Para além disso é demonstrada a importância deste tipo de software na área de segurança de infra-estruturas, informação e sistemas. O capítulo também apresenta os objectivos, o planeamento e as contribuições do trabalho.

O **Capítulo 2 – Panorâmica actual** divide-se em 3 subcapítulos: "Contexto", "Panorâmica e trabalho relacionado" e "Tipos de ataques". O primeiro subcapítulo explica o contexto em que foi desenvolvida este projecto. A secção "Panorâmica e

trabalho relacionado" descreve de uma forma mais detalhada a realidade actual existente do processo de recolha de indicadores, introduzida no subcapítulo "Motivação". O subcapítulo "Tipos de ataques" define, através de uma breve descrição, cada um dos diferentes tipos de incidentes que estão presentes na taxonomia da European Union Agency for Cybersecurity (ENISA), associados aos vários tipos de ciberataque. Estas definições permitem perceber a importância de conhecer, a priori, os indicadores de compromisso.

O **Capítulo 3 – Desenho da Zwerg** apresenta a arquitectura da solução desenvolvida. Este capítulo é dividido em 4 subcapítulos: "Estratégia", "Requisitos", "Arquitectura da Zwerg" e "Núcleo da aplicação". O primeiro subcapítulo define as estratégias que são implementadas num SOC no processo de gestão e reposta a incidentes. O segundo subcapítulo enumera todos os requisitos definidos pelo SOC do MAI. O subcapítulo "Arquitectura da Zwerg" explica o conjunto de ferramentas utilizadas no desenvolvimento da aplicação e a sua hierarquia. O "Núcleo da Aplicação" detalha em pormenor as principais interacções entre os diferentes componentes do núcleo da Zwerg.

O **Capítulo 4 – Implementação** explica o processo de implementação da Zwerg e divide-se em cinco subcapítulos: "Recolha de notícias", "Processamento de dados e expressões regulares", "Validação de indicadores", "Armazenamento de dados" e "Integração com o MISP". O primeiro subcapítulo explica o processo recolha de notícias a partir de uma lista de feeds. O segundo subcapítulo descreve os mecanismos usados no processamento de dados recolhidos. O terceiro subcapítulo explica os mecanismos de validação de indicadores usados pela Zwerg. Por fim são explicados os formatos de armazenamento de dados e o processo de integração com a plataforma MISP.

O **Capítulo 5 – Aplicações Web** está divido em 4 subcapítulos: "Objectivos", "Visualização de dados", "Consulta de informação de detalhada" e "Configuração de feeds". Neste capítulo abordamos: os principais objectivos da aplicação web; os mecanismos de visualização de dados disponíveis na aplicação; os mecanismos de consulta de informação detalhada sobre um determinado indicador; os mecanismos de configuração de feeds a partir da aplicação web.

No **Capítulo 6 – Resultados** são detalhados os testes efectuados em cada um dos módulos e os resultados obtidos. Neste capítulo abordamos também o resultado dos indicadores recolhidos no contexto do SOC do MAI.

No **Capítulo 7 – Conclusão e trabalho futuro** são apresentadas as principais conclusões após cada etapa do desenvolvimento deste projecto, bem como procedimentos a serem implementados em futuras versões.

Capítulo 2 **Panorâmica actual**

Neste capítulo abordamos o contexto do cenário actual do processo de recolha de indicadores de compromisso. Durante esta abordagem definiremos qual o estado actual das plataformas existentes no mercado e quais os problemas resolvidos pela Zwerg. Serão também explicados os vários tipos de ciberataques e a importância de ter conhecimento dos indicadores, relacionados com esses ataques, a priori.

No contexto deste projecto OSINT é toda a informação obtida a partir de fontes abertas e públicas, ou seja, informação que pode ser utilizada, directamente ou indirectamente, na análise e resolução de incidentes informáticos e consequente protecção de ameaças informáticas.

2.1 Contexto

Existem actualmente diversas aplicações de recolha de informação OSINT, essenciais no contexto de análise de incidentes num SOC. No contexto deste projecto OSINT é toda a informação, obtida a partir de fontes abertas e públicas, que pode ser utilizada, directamente ou indirectamente, na análise e resolução de incidentes informáticos e consequente na protecção contra ciberataques futuros.

O principal problema das plataformas de recolha de informação OSINT disponíveis actualmente no mercado é que são complexas e requerem, por vezes, licenciamento anual cujo custo nem sempre pode ser suportado pela administração de um SOC. Outro grande problema destas ferramentas é que nem sempre o seu conteúdo é de uma fonte fidedigna, ou seja, as suas bases de dados são maioritariamente preenchidas por utilizadores que se registam nessas plataformas, o que pode levar a que as mesmas possam conter informação incorrecta ou incompleta. Por exemplo um Internet Protocol (IP) pode pertencer a uma máquina infectada e por isso é considerado malicioso. No entanto, quando essa máquina deixa de ser maliciosa a informação sobre esse IP, que se encontra nas bases de dados de IOCs, nem sempre é actualizada fazendo com que estas bases de dados estejam continuamente a crescer e os seus dados deixem de ser exactos.

Os indicadores disponíveis nestas plataformas online são, por norma, conteúdo fixo, ou seja, quando os indicadores são adicionados a uma BD, não são mais actualizados. No entanto existem cenários em que as listas de indicadores existentes nas bases de dados das plataformas online têm de ser actualizadas, tais como vulnerabilidades que são corrigidas e IPs ou domínios que deixam de ser maliciosos. Estas actualizações evitam que as bases de dados de indicadores tenham um crescimento descontrolado e se tornem pouco eficazes no que toca ao seu aproveitamento com fonte de informação preciosa para ferramentas SIEM.

Quando estas bases de dados se tornam demasiado extensas as ferramentas de análise e correlação de *logs* tornam-se lentas, o que leva a que uma simples análise de um IP ou de um domínio se torne complexa devido à quantidade de indicadores

presente na BD. Para além disso, a existência de indicadores incorrectos nestas bases de dados pode fazer com que sejam despoletados alertas de incidentes que são falsos positivos, pois as Active Lists do SIEM, que são preenchidas com o conteúdo das bases de dados de indicadores, contêm informação incorrecta. A maioria das ferramentas de recolha de informação OSINT não actualiza com a devida frequência as listas de indicadores presentes nas suas bases de dados.

Por tudo isto é necessário um controlo preciso e eficaz das bases de dados de IOCs por forma a que as ferramentas existentes num SOC sejam eficientes em termos de tempo de análise e eficazes em relação ao conteúdo que estão a correlacionar.

Actualmente cabe aos gestores de bases de dados de IOCs controlar os indicadores presentes nessas bases de dados. De acordo com os parâmetros definidos pelo COSI um indicador deve ser considerado malicioso durante um período de seis meses após a primeira vez que é inserido na base de dados de indicadores.

2.2 Panorâmica e trabalho relacionado

A constante evolução das *frameworks* e linguagens de programação tornou mais diverso e complexo o cenário de desenvolvimento de aplicações informáticas. As *frameworks* de desenvolvimento de aplicações web estão em constante evolução, pelo que se torna difícil acompanhar a esta evolução. O número de vulnerabilidades *zero-day* encontradas actualmente é cada vez maior, pelo que é necessário um acompanhamento incisivo, por forma a evitar que estas sejam exploradas com sucesso. Vulnerabilidades *zero-day* são vulnerabilidades que foram recentemente descobertas e para as quais ainda não existe um *patch* ou uma actualização para resolver essa mesma vulnerabilidade. Por vezes estas vulnerabilidades são exploradas com sucesso pelos atacantes sem que os responsáveis, pelo ambiente atacado, possam reagir a tempo de impedir o sucesso do ataque.

A maioria das vulnerabilidades *zero-day* são anunciadas através de investigadores de segurança informática e *ethical hackers*, que recorrem a *blogs* e sites da especialidade para divulgar informação crítica que pode ser útil para os responsáveis por qualquer tipo de plataforma ou serviço informático, bem como para os analistas de um SOC.

Apesar da complexidade dos sistemas ter aumentado, tornou-se cada vez mais fácil realizar ataques informáticos, devido à grande variedade de linguagens e aplicações que suportam esses sistemas e que têm vulnerabilidades conhecidas. Por outro lado, existem cada vez mais aplicações que auxiliam o atacante nas diversas fases de um ataque informático, tais como: reconhecimento, descoberta e exploração de vulnerabilidade. Este tipo de aplicações é normalmente criado com o objectivo de auxiliar os profissionais de segurança informática. No entanto é recorrentemente utilizado pelos atacantes para efectuar reconhecimento dos ambientes informáticos ou exploração de vulnerabilidades existentes no ambiente informático alvo. Exemplos disso são o Network Mapper (Nmap), que permite avaliar a segurança da rede, ou o SqlMapper que permite detectar e explorar

vulnerabilidades em bases de dados Structured Query Language (SQL). Este tipo de ferramentas é comum na maioria dos ataques informáticos pelo que as entidades são obrigadas a fortalecer os seus perímetros de segurança através de medidas de protecção e barramento de acessos, tais como a utilização de equipamentos e software específico que impeça a realização deste tipo de ataques, como por exemplo uma Web Application Firewall (WAF).

As ferramentas actuais de recolha de informação OSINT são na sua maioria pagas e são ferramentas de largo alcance, ou seja, com elas é possível realizar um elevado número de operações de análise, como por exemplo análise de malware, de ataques de phishing, DoS/DDoS, criptomining ou ransomware, análise de domínios e subdomínios, Unique Resource Location (URL) Unique Resource Identification (URI) ou padrões de ataque, ou mesmo geolocalização. Estas ferramentas são bastante eficazes, mas, no entanto, são tipicamente dispendiosas, devido às licenças de utilização que necessitam de ser renovadas.

Existem, no entanto, ferramentas gratuitas de OSINT que podem ser usadas para recolher indicadores sobre um tipo específico de ataque, bem como informações sobre IOCs relativos a ataques que afectem software e hardware do ambiente monitorizado. Estas ferramentas providenciam ao analista uma avaliação dos indicadores sobre um determinado ataque. Esta avaliação é útil para um gestor de bases de dados de IOCs pois permite-lhe criar uma hierarquia de indicadores que lhe permite uma melhor análise.

Na procura por ferramentas que desempenham o processo de recolha e análise de informação OSINT, foi possível identificar três grandes ferramentas: Hybrid Analysis, OTX Allien Vault e OSINT+ (4IQ).

A ferramenta Hybrid Analysis é uma página web que presta um serviço de análise de malware online. A partir da página web é possível obter OSINT de uma forma gratuita, através da API que o site disponibiliza. O serviço permite obter informação OSINT sobre diferentes tipos de conteúdo que incluem, por exemplo, domínios maliciosos, campanhas de phishing, entre outros indicadores. Esta ferramenta é muito usada para fazer análise estática e dinâmica, através de *sandboxing* de malware. Contudo é uma ferramenta que é muito direccionada para o malware, não fornecendo informação útil sobre outros tipos de ataque. Para além disso, não disponibiliza uma BD actualizada de IOCs, fornecendo, no entanto, ferramentas de pesquisa individual que permitem a validação de IOCs.

A aplicação OTX AlienVault é outra das ferramentas utilizadas em ambientes de SOC. Consiste numa plataforma online onde utilizadores registados conseguem inserir e partilhar os IOCs que encontram na execução de tarefas de análise, por isso a BD desta ferramenta contém grande quantidade de IOCs nestas tarefas. Esta ferramenta disponibiliza uma API de integração com outras ferramentas de recolha e gestão de IOCs existentes no mercado, como por exemplo o MISP, o que torna o processo de inserção de IOC's na BD completamente automático. No OTX AlienVault existem grupos de utilizadores que publicam IOCs relativos a temas específicos, o que facilita a selecção de IOCs presentes em toda a BD. Um gestor de um SOC pode seguir apenas os grupos de utilizadores referentes a uma determinada área, por

exemplo, malware. Assim o gestor de BD apenas recebe no MISP IOCs relacionados a um tema específico, nomeadamente IOCs relativos a ataques que envolvem activos existentes no ambiente que está a monitorizar. Contudo, como a ferramenta é de livre utilização, qualquer utilizador pode inserir IOCs incorrectos, o que torna a fiabilidade da ferramenta dependente dos utilizadores que mantêm a bases de dados. Outra desvantagem desta aplicação prende-se com facto de a maioria das listas de IOCs presentes na BD não serem actualizadas regularmente. Isto faz com que os indicadores que deixam de ser válidos, como por exemplo IPs das máquinas que estiveram infectadas e que já não são maliciosas, não são removidos das listas existentes, fazendo com que existam sempre indicadores falsos positivos nestas listas.

A ferramenta OSINT+ não é de utilização gratuita, pelo que nesta análise não foi possível estabelecer a sua abrangência, nem a eficácia dos resultados produzidos.

Foi no âmbito de combate a estas falhas na fiabilidade de recolha de IOCs, bem como os custos associados à utilização de ferramentas pagas, que surgiu a necessidade de criar um software especializado na recolha e tratamento de IOCs. A sua implementação no contexto do SOC do MAI veio permitir obtenção de resultados práticos que determinaram valor acrescentado por este software na recolha e tratamento dos indicadores de acordo com as normas definidas pelo COSI.

2.3 Tipos de ataques

Nesta secção iremos explicar os ataques informáticos mais comuns que estão classificados como incidentes na taxonomia oficial da ENISA. Desta forma pretendemos evidenciar a importância dos indicadores relativos a cada um destes ataques.

Actualmente existem inúmeros tipos de ataques informáticos que podem ser classificados com base em diferentes características e vectores de ataque. Segundo a taxonomia oficial da ENISA a taxonomia de classificação de incidentes de segurança reportados por uma equipa de Computer Security Incident Response Team (CSIRT) divide-se em 11 categorias*:* abusive content, availability , information gathering, malicious code, intrusion attempts, intrusions, information security content, fraud, vulnerable, other e test [9]. Cada uma das categorias divide-se em várias subcategorias mais específicas e é associada a um ou vários tipos de ataque informático.

Nos dias de hoje as categorias de ataques mais comuns são: abusive content, information gathering, availability e malicious code.

Existem diversos tipos de código malicioso (malicious code) e segundo a taxomia de incidentes definida pela ENISA, esta categoria de ataques engloba: vírus, worm, trojan, spyware, dialler e rootkit.

Foi no final da década de 80 que surgiu uma amostra daquilo que se viria a tornar uma das grandes ameaças ao mundo digital actual: o malware. Existem vários

tipos de malware que têm diferentes objectivos. Entre os mais conhecidos encontramos o worm, o trojan, o vírus e, numa história mais recente, o ransomware.

Um dos primeiros exemplos de malware remonta a 1988, quando o professor Robert Tappan Morris criou um worm que tinha apenas o objectivo de saber qual é o "tamanho" da Internet, mas depressa se disseminou por redes desconhecidas causando negação de serviço a milhares de computadores, devido a um erro de programação que não validava se existia um processo deste worm em execução na máquina a infectar. Como consequência, a máquina infectada executava um processo cada vez que o worm se tentava disseminar, até esgotar os seus recursos computacionais. Nesse ano o Government Accountability Office do governo Americano estimou que o dano causado pelo worm ao governo Americano rondava um valor entre 100.000$ e 10.000.000$ [10]. A descoberta deste worm revelou ao mundo o poder do impacto que uma pequena porção de código pode causar.

A cobertura mediática do ataque do worm de Morris foi uma das causas do desenvolvimento de novos worms e novos tipos de ataques informáticos, que permitiam causar danos incalculáveis com relativamente pouco esforço e uma grande recompensa, visto que a segurança dos sistemas informáticos era, até à altura, algo teórico e sem grande importância e investimento.

Outro tipo de malware que surgiu na sequência de um aumento cada vez maior do número de utilizadores de Internet foi o adware, cujo objectivo é exibir grandes quantidades de publicidade indesejada ao utilizador. Este tipo de serviços é por vezes contratado por muitas empresas para que possam chegar ao maior número de utilizadores possível, mesmo que de forma intrusiva e abusiva. Provou-se ser uma estratégia com resultados práticos, pois muitos utilizadores clicam na publicidade e geram receitas lucrativas para as empresas. Por outro lado, esta publicidade pode ser usada por hackers para esconder *links* que reencaminham o utilizador para domínios maliciosos infectando a sua máquina. Apesar de existirem cada vez mais ferramentas exclusivas no tratamento deste tipo de ataque, os criadores deste malware encontram sempre uma nova forma de contornar essas barreiras e fazer chegar ao utilizador final publicidade indesejada. No entanto, as consequências para a máquina do utilizador final nem sempre são malignas porque muitas vezes o utilizador não chega a ficar infectado.

Existem, no entanto, outros tipos de malware mais poderosos e que marcaram a história dos ataques informáticos pelo impacto financeiro e mediático que causaram. O trojan, por exemplo é um tipo de malware que se instala na máquina do utilizador infectando-a e efectuando ataques remotos a partir da mesma, mas ao contrário de outros tipos de malware, não tem como objectivo a auto-replicação para outras máquinas. O trojan é capaz de criar backdoors para que um atacante se possa conectar à máquina da vítima, sem que esta se aperceba.

O spyware é um tipo de malware que tem como principal objectivo recolher dados da vítima. O spyware é instalado em equipamento da vítima e recolhe informação confidencial. As quatro formas mais comuns de fazer chegar o *spyware* ao equipamento da vítima são: através de um trojan, *third party programs*, *tracking cookies* e *system monitor tools*.

O dialler é um tipo de malware menos comum nos dias de hoje, sendo cada vez mais raro a sua detecção em incidentes de um SOC.

O rootkit é um software que tem como objectivo fazer com que o atacante obtenha privilégios administrativos sobre o equipamento da vítima. Rootkit é a junção das palavras root e kit que no linux são palavras utilizadas para designar administrador de sistema e software desenhado para obter privilégios de administração sem o conhecimento do utilizador. Este tipo de malware é extremamente difícil de detectar e normalmente é constituído por três componentes: dropper, loader e o rootkit. O dropper é normalmente um programa executável ou um ficheiro que instala o rootkit. É comum encontrar este tipo de malware em ficheiros Portable Document Format (PDF) ou documentos em formato Microsoft Word que contêm Open Actions ou código JavaScript. Outra forma comum é através do uso de ficheiros legítimos que foram corrompidos pelo atacante por forma a incluir um dropper sem que o utilizador tenha conhecimento.

Actualmente a grande maioria dos utilizadores de serviços na Internet possui um endereço de correio electrónico (email), que é um dos serviços mais antigos na Internet e tem perdurado ao longo dos anos. O email atingiu actualmente um papel essencial na troca de informação e tornou-se cada vez mais inteligente e seguro. Por dia são trocados cerca de 220 milhões de emails através de serviços gratuitos de email disponíveis na Internet. Como estes serviços são gratuitos um utilizador sem muitos conhecimentos informáticos pode criar um endereço de email, o que leva a que se torne um alvo fácil de campanhas de phishing. Estas são uma prática fraudulenta que consiste no envio de emails pelo atacante fazendo-se passar por uma empresa legítima com o objectivo de obter dados pessoais ou corporativos. Em 2017 o impacto deste vector de ataque foi tão grande que 71% dos ciberataques teve origem no phishing. A presença de emails de phishing e spear phishing, que é uma forma de phishing direccionado a um utilizador alvo, no correio electrónico é constante e tem vindo a crescer (2% de 2016 para 2017), pois tornou-se uma forma fácil de obter credenciais da vítima[8].

Uma das consequências dos ataques de phishing é a disseminação de malware do tipo vírus. O vírus infecta normalmente os computadores da vítima através do download de ficheiros presentes em emails de phishing ou acessos a domínios maliciosos por parte do utilizador, e esconde-se nos ficheiros da máquina da vítima. Para não serem detectados pelo software de detecção de ameaças os vírus por vezes alteram os dados do tamanho do ficheiro infectado e datas da sua modificação, para passarem despercebidos. Quando estes se instalam na máquina da vítima executam o seu código malicioso e replicam-se de forma a atingir o maior número possível de máquinas.

O abusive content é composto por três tipos de incidentes: spam, harmfull speech e child/sexual/violence content. O spam é definido como sendo o envio em massa de emails não solicitados pelo destinatário. Tipicamente estas mensagens são idênticas e enviadas para um conjunto de utilizadores sem o conhecimento prévio dos mesmos e o seu conteúdo é normalmente malicioso ou abusivo. O harmfull speech consiste no cyber stalking, racismo ou ameaças contra um ou mais

indivíduos. O child/sexual/violence content inclui pornografia infantil e todo o conteúdo classificado como sendo violento ou de cariz sexual, como por exemplo vídeo ou imagens.

A recolha de informação (information gathering em inglês) representa um dos incidentes mais comuns num SOC porque reflecte a etapa inicial de um ataque informático: o reconhecimento do alvo. Esta categoria de incidente divide-se em 3 secções: scanning, sniffing e social engeneering.

Scanning é a classe de incidentes que engloba vários tipos de ataque, como por exemplo port scans e directory transversal. Este tipo de ataques de reconhecimento activo é muito utilizado pelos atacantes para explorar possíveis portos abertos no servidor ou a sua hierarquia de directorias. O sniffing tem como objectivo a obtenção ilegal de dados, através da sua escuta, transmitidos na rede. Esta informação pode ser depois usada para fazer reconhecimento, obter credenciais ou informações sobre os métodos de autenticação. A secção de incidentes de social engineering inclui incidentes que envolvem a manipulação do utilizador de forma indevida, sem recorrer directamente a sistemas informáticos, com o objectivo de obter informações que não são públicas.

Ataques à disponibilidade de um serviço têm como objectivo inundar o servidor de pedidos até atingir o limite de maneira a que este não consiga responder a pedidos legítimos. Esta categoria divide-se em 4 secções: Denial of Service (DOS), Distributed Denial of Service (DDOS), sabotage e outage.

O DOS é um ataque em que o servidor é bombardeado com inúmeros pedidos em simultâneo de forma que o servidor atinja a sua capacidade máxima de resposta a pedidos e fique assim impossibilitado de responder a pedidos legítimos. Ao receber centenas ou milhares de pedidos de ligação o servidor mantém as ligações activas e quando atinge o limite rejeita pedidos de novas ligações. Exemplos mais comuns deste ataque são o SYN flood e o ICMP flood. Estes dois ataques exploram o facto de as ligações entre o servidor e o cliente serem efectuadas em múltiplos passos. Os clientes maliciosos, hackers, efectuam pedidos de ligação ao servidor e quando recebem a resposta não continuam com o processo de estabelecimento da ligação. O servidor tem definido no seu protocolo um tempo máximo de espera para estabelecimento da ligação, visto que por vezes os pacotes se podem atrasar na rede. Como o servidor tem um número máximo de ligações que pode ter em simultâneo, o atacante efectua muitos pedidos para que o servidor deixe de responder a pedidos legítimos.

O DDOS tem o mesmo princípio de funcionamento, mas ao invés de ser apenas uma máquina maliciosa a efectuar os pedidos para um servidor estes são realizados por um conjunto de máquinas ligadas em rede que geram milhares de pedidos em simultâneo comandados pelo atacante. Um DDOS permite efectuar ataques a servidores com maior capacidade e permite que esses ataques não sejam tão facilmente detectados porque partem de vários endereços IP.

O ataque sabotage leva a que o atacante consiga sabotar o sistema e fazer com que ele deixe de prestar serviços no seu modo normal de funcionamento, fazendo

com que o atacante passe a controlar esses mesmos serviços. Este tipo de ataque pode ter um impacto financeiro grande para entidade caso não seja mitigado no período imediato à sua detecção.

O outage é um tipo de incidente em que o atacante consegue fazer com que os serviços atacados deixem de funcionar, quer seja durante um período de tempo quer seja indefinidamente.

O mais importante é que o impacto da classe de ataques malicious code tem vindo a aumentar nos últimos anos como principal consequência da evolução tecnológica que temos vindo a assistir.

A variedade de ciberataques tendo vindo a crescer nos últimos anos, resultado da evolução dos ambientes informáticos e do contante aumento da troca de dados através de serviços disponibilizados na Internet. Tendo em conta o aumento da quantidade e da variedade destes ataques torna-se clara a necessidade das organizações se precaverem contra este tipo de ataques. Neste capítulo o papel das organizações começa a ser cada vez mais proactivo com o objectivo de impedir o sucesso destes ataques descritos.

Capítulo 3 **Desenho da Zwerg**

Este capítulo explica a nossa abordagem na definição da arquitectura da Zwerg de acordo com os objectivos a que nos propusemos na construção deste software e os requisitos definidos pelo SOC do MAI. O capítulo está dividido em 3 subcapítulos: Estratégia, Requisitos e Arquitectura da Zwerg.

No subcapítulo "Estratégia" explicamos o modo de funcionamento e os diferentes tipos de estratégias implementadas num SOC, e a importância de uma ferramenta como a Zwerg no SOC de uma instituição como o MAI. No subcapítulo "Requisitos" iremos enumerar os principais requisitos definidos pelo MAI. Por fim, no subcapítulo "Arquitectura da Zwerg", iremos explicar como foi elaborada a estrutura deste software, comparando a arquitectura que foi desenvolvida na fase embrionária do projecto com a arquitectura final.

3.1 Estratégia

As empresas que fornecem serviços para terceiros ou que tenham uma interface de acessos aos seus serviços disponível para utilizadores da Internet são o tipo de empresas alvo da maioria dos ataques informáticos. Estas organizações tentam hoje proteger-se contra estes ataques recorrendo a dois tipos de estratégias: preventiva e reactiva.

A estratégia preventiva consiste na implementação de medidas que têm como objectivo mitigar os ataques informáticos, bem como impedir o acesso dos atacantes a informação confidencial, o acesso a parâmetros de configurações do sistema ou, em casos mais extremos, acesso ao controlo do sistema com privilégios de administração. Esta estratégia pode passar pela instalação de software, como por exemplo antivírus e firewalls, ou por confirmações de controlo de acesso ao sistema por parte dos utilizadores.

A estratégia reactiva é a acção realizada pela organização como resposta a um ataque informático. A grande maioria dos ciberataques são difíceis de mitigar numa fase inicial porque, apesar de serem na sua grande maioria conhecidos, os padrões de ataque e os indicadores são ainda desconhecidos pelo utilizador comum. Por isso, as empresas recorrem-se de equipas especializadas em segurança informática para manter os seus sistemas informáticos seguros.

As equipas de um SOC são responsáveis por monitorizar todos os eventos e actividade da rede num ambiente informático de uma organização. Estas equipas trabalham normalmente com uma plataforma chamada SIEM que é um sistema de agregação, gestão e correlação de dados, baseados em eventos, que facilita a detecção e análise de comportamentos anómalos num sistema. Para obter estes dados a plataforma SIEM usa conectores, ou agentes, que estão instalados nos principais componentes da rede: firewalls, routers, servidores de Domain Name System (DNS), entre outros. Estes conectores recolhem os *logs* de outras

ferramentas, como é o caso dos Intrusion Prevention Systems (IPS) e dos Intrusion Detections Systems (IDS). Após a recolha e normalização dos *logs* por parte dos agentes cabe à ferramenta SIEM a correlação de toda a informação de acordo com as regras definidas. Este software contém um conjunto de regras que actuam sobre os *logs* e quando as regras são despoletadas geram um alerta que informa o analista sobre um possível incidente informático no sistema. Estes eventos são os relatórios de incidente que são gerados aquando da detecção de padrões de tráfego anómalo no sistema. Estes incidentes podem ser resultado de tráfego anormal no sistema proveniente de acções benévolas de utilizadores, ou de tráfego anormal proveniente de acções maliciosas de utilizadores mal-intencionados. Os relatórios contêm informação útil para que o analista consiga recolher evidências sobre o comportamento descrito no relatório de incidente e à posteriori identificar se estas acções têm ou não objectivos maliciosos.

Por forma a melhorar a eficácia das ferramentas usadas num SOC é necessário ter uma atitude proactiva, ou seja, não só reagir a eventos e alertas despoletados pelo SIEM, como também prevenir futuros ataques com novas regras baseadas em IOCs recolhidos.

A prevenção é uma das formas mais eficazes de mitigar o impacto causado por novos tipos de ataque, bem como a constante adaptação das plataformas que monitorizam os sistemas de larga escala [6]. Nos dias de hoje é possível obter grandes quantidades de informação sobre ataques informáticos de uma forma rápida e de acesso gratuito através da Internet.

É neste contexto que a ferramenta Zwerg vem ajudar os analistas de um SOC a obter informação de uma forma mais incisiva, objectiva e concreta sobre as principais tendências dos ataques informáticos no mundo actual. A ferramenta Zwerg recolhe informação sobre ataques informáticos em artigos online, processa-a e analisa-a de forma a que esta seja fornecida ao analista de forma mais precisa e objectiva. A informação recolhida em bruto é processada de acordo com os parâmetros definidos nas configurações de forma a reduzir substancialmente o espectro de ataques informáticos para que a Zwerg possa realizar uma análise objectiva em tempo real.

O resultado desta análise é metainformação sobre os indicadores associados aos principais ataques, vulnerabilidades, entre outros tópicos, que podem ser utilizados pelo analista e pelo gestor de bases de dados e IOCs por forma a manter o ambiente monitorizado pelo SIEM seguro. Desta forma poderá ser possível prevenir alguns dos ataques anteriormente descritos, através de bloqueios ou recomendações informativas, durante o seu período inicial de propagação, como por exemplo a fase em que o atacante faz o reconhecimento dos equipamentos de perímetro.

3.2 Requisitos

Os requisitos definidos pelo MAI na construção deste software tiveram em conta o modo actual de funcionamento do SOC do MAI e os recursos disponíveis para o desenvolvimento da Zwerg. A principal finalidade destes requisitos foi definir um

conjunto de objectivos que deveriam ser cumpridos aquando do término do período relativo ao desenvolvimento deste projecto.

No processo de construção do software que nos propusemos a desenvolver neste projecto foram estabelecidos diversos requisitos funcionais.

Requisitos funcionais:

- A Zwerg deve recolher três tipos de indicadores diferentes: IP, domínio/URL e hash;
- A Zwerg deve ser capaz de armazenar os IOCs numa estrutura de dados definida pelo MAI que inclui: data da recolha do IOC, URL da notícia onde se encontra o IOC, resposta obtida através das aplicações externas e o nível de criticidade do IOC definido pelo MAI;
- O sistema deve ser capaz de classificar cada um dos IOCs encontrados de acordo com o nível de criticidade definido pelo MAI, sendo estes:
  - SAFE,
  - WARNING,
  - SUSPICIOUS,
  - CRITICAL;
- A Zwerg deve ter capacidade de criar eventos na plataforma MISP para armazenar os IOCs encontrados;
- Os eventos criados no MISP devem ser organizados por data e tipo de IOC;
- Cada IOC introduzido na plataforma MISP deve ser associado como atributo a um evento;
- Cada IOC deve ter um conjunto de etiquetas (tags em Inglês) associados;
- Cada IOC inserido no evento do MISP deve ter associados um nível de criticidade obtido através de aplicações externas;
- A Zwerg deve ter uma interface web que permita:
  - Apresentação resumida dos IOCs e metainformação recolhida sobre eles para visualização num *videowall*,
  - Sistema de pesquisa de IOCs para obtenção de informação mais detalhada,
  - Sistema de configuração dos endereços de *feeds*,
  - Sistema de pesquisa de notícias por palavra chave, e
  - Visualização dos gráficos sobre os IOCs recolhidos.

Os requisitos não funcionais estão divididos nas seguintes categorias: armazenamento, custo, eficácia e usabilidade.

Requisitos de armazenamento:

- Não exceder 10GB de armazenamento de dados;
- As notícias recolhidas a partir dos *feeds* devem ser armazenadas numa estrutura de dados definida pelo COSI.

Requisitos de custo:

- A Zwerg deve utilizar aplicações externas recorrendo apenas a planos gratuitos.

Requisitos de eficácia:

- A aplicação web deve ser regularmente actualizada para permitir a visualização de IOCs existentes recentemente recolhidos.
- A Zwerg deve efectuar a recolha e processamento de indicadores duas vezes por dia.

Requisitos de usabilidade

- A interface web deve ser interactiva e de fácil utilização.
- A aplicação web deve usar um esquema de cores semelhante ao esquema utilizado pelas aplicações presentes no videowall.

3.3 Arquitectura da Zwerg

A *framework* definida para a implementação, bem como a hierarquia de processos, ilustradas na Figura 3.1, sofreram poucas alterações em comparação com o resultado final, apresentado na Figura 3.2. Em função da evolução do processo de implementação da aplicação surgiram novos requisitos, por parte do COSI, relativos a novas funcionalidades. A avaliação dos indicadores recolhidos através de aplicações externas foi um dos requisitos que surgiu durante o processo de implementação, para responder a possíveis falsos positivos.

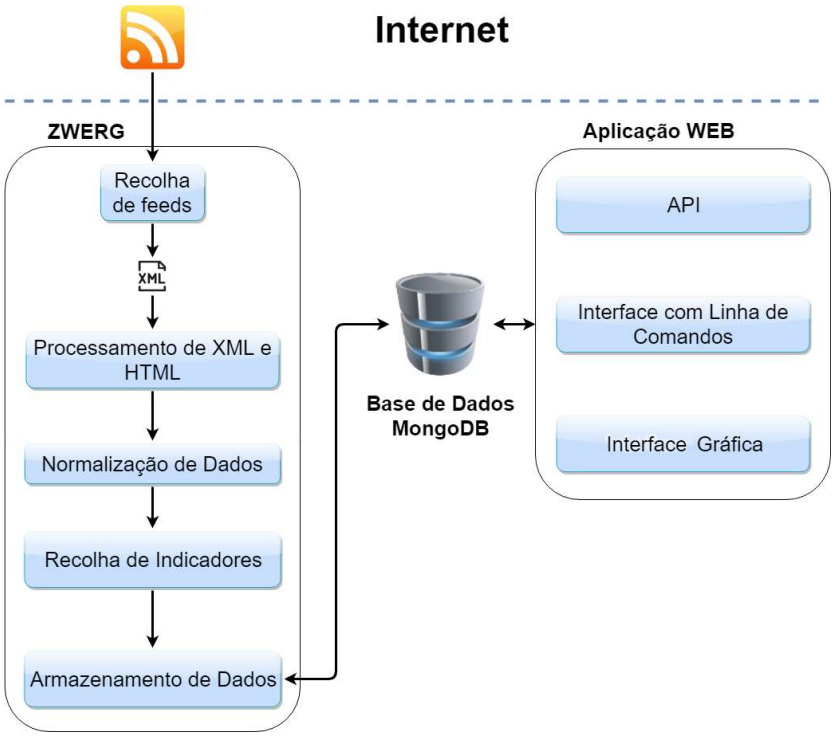

Figura 3.1 - Arquitectura inicial da solução.

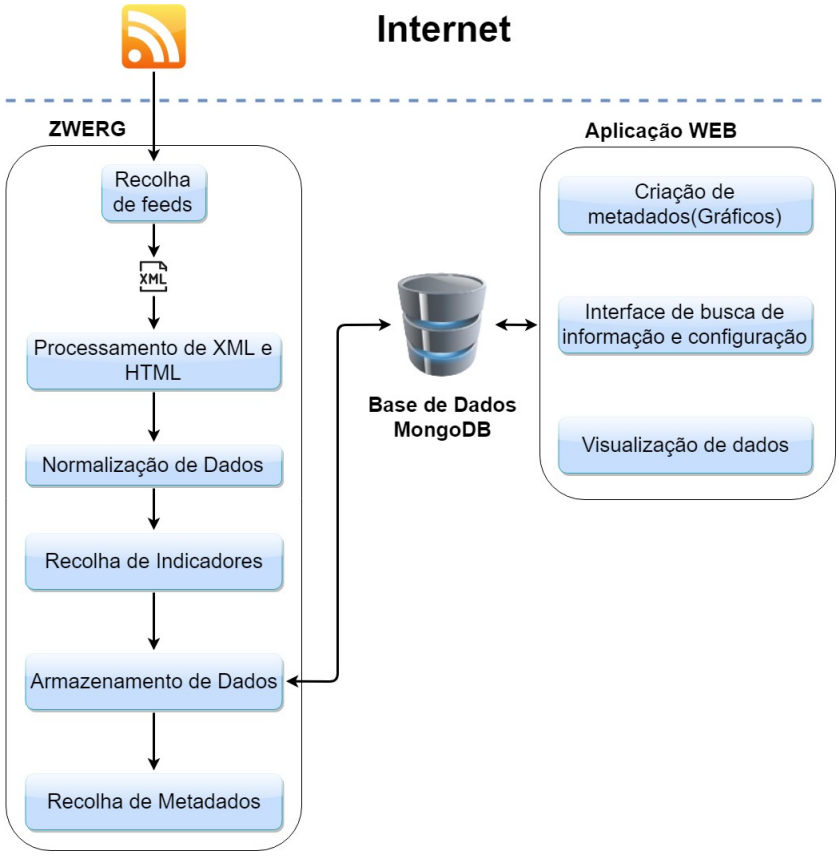

Figura 3.2 - Arquitectura final da solução.

Outra alteração relevante teve impacto na aplicação web, que deixou de ter uma interface com linha de comandos integrada para passar a ter todas as funcionalidades e configurações a partir de interfaces gráficas disponibilizadas nas várias páginas da aplicação, a pedido do COSI.

A interface com linha de comandos que estava inicialmente definida como sendo uma interface de manipulação de dados e configurações entre o software e o utilizador final não foi implementada porque os mecanismos de configuração e a interacção entre a aplicação e o utilizador terem sido integrados na aplicação web. Desta forma o utilizador não precisa de ter conhecimento de qualquer tipo de sintaxe para interagir com a Zwerg, pois a aplicação web é interactiva e de fácil utilização.

Na fase embrionária do projecto foi pensada uma API para que fosse possível aceder à informação colectada pela Zwerg através de outras aplicações e serviços. No entanto, após ter sido criada a base de dados MISP com os indicadores e os respectivos metadados associados deixou de existir a necessidade de desenvolver essa API. O MISP permite exportar os dados existentes na BD através da interface gráfica ou através da biblioteca PyMisp, levando a API que pretendíamos criar deixe de ter relevância para o projecto.

Durante o processo de desenvolvimento surgiram requisitos relacionados com a visualização de metadados e busca de informação. O requisito de visualização de metadados obriga a que seja possível visualizar o nível de criticidade de cada indicador na página inicial.

Para satisfazer a visualização de metadados foi criada uma página de gráficos que permite agregar os IPs recolhidos pela Zwerg por forma extrair informação sobre a geolocalização dos IPs atacantes.

Toda a restante estrutura que estava inicialmente definida manteve-se inalterada.

Capítulo 4 **Implementação**

Este capítulo explica de uma forma genérica as abordagens que decidimos utilizar nos vários desafios encontrados no desenvolvimento da solução final e aborda os detalhes técnicos de forma clara e concisa para que seja possível perceber as razões por detrás das escolhas efectuadas, bem como a forma como foram implementadas. Ao longo deste capítulo iremos abordar os vários standards de feeds que existem, bem como o processo de recolha de dados definido para a Zwerg; as estruturas de dados definidas para armazenar toda a informação, tais como notícias, indicadores de compromisso e metadados; a implementação das expressões regulares na recolha de dados; os principais métodos utilizados para validar os indicadores recolhidos usando plataformas externas à Zwerg; e, por fim, a implementação do módulo de integração com a plataforma MISP.

O núcleo da aplicação é composto por vários módulos, escritos em Python que são responsáveis pelo processamento e armazenamento da informação. Cada um dos módulos executa um conjunto de tarefas que varia entre a recolha das notícias a partir dos *feeds* Extensible Markup Language (XML), o armazenamento dessa informação, a normalização dos dados, a extracção de informação a criação de metadados, entre outras tarefas.

Para desenvolver cada um dos módulos foi necessário definir a linguagem base de desenvolvimento dos módulos. A escolha da linguagem de programação Python foi feita com base em diversos factores: a existência de bibliotecas de Python actualizadas que permitem a interacção com os diversas softwares necessários, tais como o MongoDB e o MISP; a experiência e o conhecimento prévio da linguagem detidos pelos utilizadores da Zwerg, bem como para que futuros ajustes possam ser concretizados sem a presença de suporte técnico; a simplicidade na criação de novas funcionalidades de acordo com a necessidade do ambiente em que a Zwerg está inserida, e a simplicidade e flexibilidade da linguagem Python.

As principais funcionalidades da Zwerg foram desenvolvidas em vários módulos de Python: zavot.py, mongo.py, indicators.py, domainVT.py, hashVT.py e importToMisp.py. A divisão em diversos módulos foi delineada ao longo do desenvolvimento do projecto e teve como objectivo a modularização do software por forma a tornar o código mais eficiente e reutilizável, bem como delinear com maior exactidão o planeamento das diversas iterações de desenvolvimento do projecto.

Na Figura 4.1 podemos observar a relação entre os diversos módulos e ter uma ideia concreta do modo de funcionamento da aplicação.

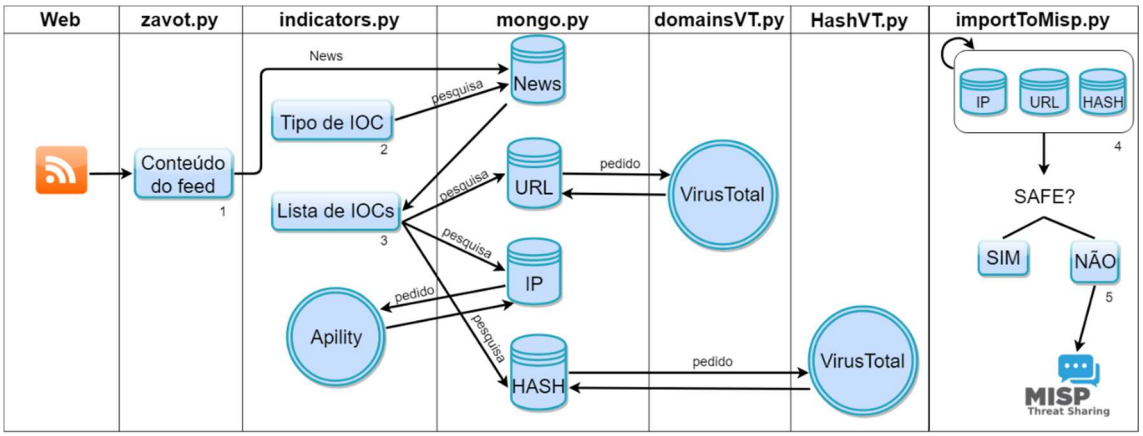

Figura 4.1 - Diagrama de acções.

O zavot.py é o módulo que faz arrancar o processo de recolha de informação a partir dos *feeds* XML que estão listados no ficheiro de configuração /config/feeds.txt. No início do processo são recolhidas todas as notícias que estão nos ficheiros de texto disponibilizados por cada um dos *links* no ficheiro feeds.txt. Para efectuar o download destes conteúdos é necessário possuir uma ligação à Internet. Após ser efectuada a recolha das notícias cada uma delas é processada individualmente para que estas sejam armazenadas de acordo com o esquema utilizado nas bases de dados de notícias. Por fim são feitas as validações necessárias por forma a não ter notícias duplicadas na BD ou documentos com formatos incorrectos, como por exemplo campos vazios. A identificação das notícias é feita através do URL que deve ser único para cada notícia.

O módulo domainVT.py é responsável por fazer os pedidos à API da aplicação VirusTotal. Este módulo recebe como input um domínio/URL proveniente do módulo indicators.py, efectua um pedido à API do VirusTotal e armazena a resposta na BD desse indicador. O resultado desta resposta é posteriormente analisado no módulo indicators.py. Desta análise resulta o nível de criticidade do indicador.

O módulo hashVT.py tem um comportamento semelhante ao módulo domainVT.py na medida em que também utiliza a API do VirusTotal. Os pedidos feitos pelo módulo hashVT.py são, no entanto, relativos ao IOC do tipo hash.

O módulo indicators.py é onde se contentram a maioria das funcionalidades da Zwerg. É neste módulo que são realizadas: a pesquisa de indicadores presentes nas notícias com recurso a expressões regulares; o armazenamento dos indicadores nas bases de dados respectivas e a normalização da estrutura de dados de cada objecto da BD.

O módulo importToMisp.py é responsável por criar todas as entradas na base de dados do MISP.

Ao longo das próximas secções iremos detalhar cada uma das acções anteriormente enumeradas.

4.1 Recolha de notícias

Really Simple Syndication (RSS) é um formato de distribuição de conteúdos online [11]. Um *feed* RSS permite que qualquer utilizador obtenha uma versão estruturada de diversos tipos de conteúdos disponibilizados online, tais como notícias, *podcasts*, *links* para ficheiros *torrent*, entre outros. A utilização de *feeds* XML é bastante comum nos dias de hoje porque permite o rápido acesso a informação, quer seja através de um URL de *feed* que disponibiliza os dados estandardizados, quere seja através de aplicações de agregação de *feeds*, tais como o Feedly [12] ou o NewsBLur [13]. Existem três versões de RSS, sendo que neste projecto optamos por usar *feeds* com a versão mais recente, RSS 2.0.

O RSS tornou a distribuição de conteúdos online standard e é caracterizado por texto formatado em XML, com uma estrutura hierárquica de etiquetas [11]. Esta estrutura pode ser observada na Figura 4.2.

```
<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0">
<channel>
 <title>RSS Title</title>
 <description>This is an example of an RSS feed</description>
 <link>http://www.example.com/main.html</link>
 <lastBuildDate>Mon, 06 Sep 2010 00:01:00 +0000 </lastBuildDate>
 <pubDate>Sun, 06 Sep 2009 16:20:00 +0000</pubDate>
 <ttl>1800</ttl>

 <item>
  <title>Example entry</title>
  <description>Here is some text containing an interesting description.</description>
  <link>http://www.example.com/blog/post/1</link>
  <guid isPermaLink="false">7bd204c6-1655-4c27-aeee-53f933c5395f</guid>
  <pubDate>Sun, 06 Sep 2009 16:20:00 +0000</pubDate>
 </item>

</channel>
</rss>
```

Figura 4.2 - Estrutura de um *feed* RSS 2.0.

Existem muitos outros formatos de distribuição de conteúdos online semelhantes ao RSS. No entanto, optamos por usar apenas o RSS porque é o standard mais amplamente usado e porque o desenvolvimento de interpretadores de novos formatos iria exceder o tempo necessário ao desenvolvimento e conclusão deste projecto em tempo útil. Durante o processo de investigação não encontramos nenhuma aplicação que usasse outro standard em prol do RSS, como por exemplo o Atom.

Apesar de o processo de desenvolvimento do interpretador de ficheiros XML se ter revelado um processo complexo, consideramos ter sido a escolha acertada não desenvolver interpretadores para outros formatos.

No início do processo de recolha de dados é carregado para o módulo zavot.py o ficheiro "\config\feeds.txt". O módulo zavot.py itera sobre cada um dos *links*. Em cada iteração este efectua o download do conteúdo do *feed* em formato XML, seguindo o standard RSS 2.0. Os *feeds* utilizados durante o processo de desenvolvimento estão listados no Anexo B. Após obter o conteúdo de todos os *links*

é necessário efectuar o processamento de cada *feed* para obter os seus dados, isto é, o URL, a descrição, o conteúdo, a data de publicação e o título. Todos os outros campos existentes no *feed* não são armazenados na BD por serem irrelevantes no âmbito da recolha de indicadores.

O processo de recolha de dados a partir dos *feeds* XML é constituído por várias fases, descritas em seguida.

Inicialmente o conteúdo é recolhido do item presente no *feed* de notícias em formato HTML. Por forma a efectuar uma pesquisa mais eficiente de indicadores na notícia, o conteúdo HTML é processado através de uma classe definida como MLStripper, que permite remover todas as etiquetas HTML presentes nos dados para que apenas o texto da notícia seja armazenado.

O RSS permite que a Zwerg consiga processar o conteúdo de notícias através do uso da biblioteca *feefparser.py*. Esta biblioteca permite converter ficheiros de *feeds* com o formato XML em dicionários de pares chave valor. Cada *feed* é assim transformado num dicionário que contem vários pares chave valor correspondentes às etiquetas e conteúdo, respectivamente, do ficheiro XML original. Com os dados armazenados numa estrutura de dados Python manipulável é possível processar a informação recolhida e criar metadados sobre ela. Após ter sido realizado o *parsing* da informação presente nos vários pares chave valor é necessário agregar os dados de cada item e armazená-los na BD dentro da colecção news. O armazenamento deste conteúdo personalizado é feito através do módulo mongo.py, desenvolvido em específico para o armazenamento dos diversos conjuntos de dados.

Os dados armazenados são constituídos por várias partes da cada uma das notícias inicialmente presentes no *feed*. Por cada notícia do *feed* é criado um objecto na BD que é composto por 6 entradas: objectId, title, url, pubDate, description e content. Cada uma destas entradas contém respectivamente: um identificador único de cada objecto na BD; título da notícia; URL da notícia; a data em que foi publicada; a descrição da notícia; e o conteúdo ou corpo da notícia.

As notícias existentes na BD são actualizadas sempre para a versão mais recente. Sempre que o módulo zavot.py é executado verifica se já existe uma notícia com o mesmo URL e no caso positivo todos os seus constituintes, ou seja, o título, o conteúdo, a descrição, entre outros, são actualizados com os novos dados recolhidos. A decisão de sobrepor os dados prende-se com a constante actualização das notícias presentes nos *feeds*. Sempre que um autor adiciona conteúdo a uma notícia e essa alteração se reflecte no *feed*, o objecto da notícia que existe na BD com o mesmo URL é actualizado. O URL foi o identificador único externo escolhido porque representa a única forma de acesso ao conteúdo da notícia que está presente na página web.

4.2 Processamento de dados e expressões regulares

Após serem recolhidas as notícias presentes nos *feeds* é necessário proceder ao processamento de toda a informação recolhida. O objectivo desta fase é recolher todos os indicadores de compromisso válidos e toda informação útil associada a

cada um deles. Sempre que um indicador é encontrado no conteúdo de uma notícia é associada a esse IOC uma lista de palavras chave que sejam encontradas na notícia a partir da qual o indicador foi recolhido. Esta lista de palavras chave é extremamente importante para uma correcta caracterização do indicador.

Durante o processo de investigação decidimos que a melhor forma de obter informação concreta e precisa a partir dos blocos de texto das notícias é através do uso de expressões regulares. Expressão regular é um conjunto de caracteres que permite pesquisar padrões num texto ou sequência de caracteres. O uso de expressões regulares é muito comum em motores de busca, processadores de texto e é inclusive utilizado em linguagens de programação através de bibliotecas disponibilizadas para o efeito. Nas próximas três secções iremos explicar, em detalhe, cada um dos *tokens* de expressão regular usado para obter o indicador a partir do conteúdo da notícia, sendo estes: IP, domínio/URL e hash.

4.2.1 Token para recolha de IPs

Para procurar informação em notícias, através de expressões regulares, é necessário definir um *token* que reflicta o padrão que procuramos. Por exemplo, quando efectuamos uma busca pelo IOC IP, é necessário procurar por conjuntos de caracteres que correspondam ao padrão definido pelos standards do protocolo IPv4, sendo este formado por 4 números, separados por caractere ponto (x.x.x.x), sendo que 'x' corresponde a um valor entre 0 e 255 inclusive [14]. Na busca pelo IOC IP definimos o seguinte *token*, exemplificado na Figura 4.3.

```
((25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)(\.|\[\.\])(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)(\.|\[\.\])(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)(\.|\[\.\])(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?))
```

Figura 4.3 - Token utilizado na busca de IPs.

A razão pela qual optamos por não definir um *token* para IPv6 foi devido á utilização pouco recorrente deste formato. Por forma a obter apenas IPs válidos no contexto de IOC, foi efectuada uma validação extra para remover os IPs 0.0.0.0, 255.255.255.255, 10.0.0.0, 10.255.255.255, 192.168.0.0, 192.168.255.255, e 127.0.0.0.

Para que estes *tokens* possam ser interpretados existem diversos motores que interpretam estas sequências de caracteres e fazem a correspondente busca pelo padrão que desejamos encontrar. Os *tokens* devem ser definidos de acordo com os padrões do motor de expressões regulares que estamos a utilizar. Neste projecto decidimos usar a sintaxe do motor de expressões regulares Perl Compatible Regular Expressions (PCRE) nas várias expressões regulares, por ser o motor compatível com Perl utilizado pelo MongoDB, bem como um dos mais utilizados actualmente [15].

4.2.2 Token para recolha de domínios/URLs

Para a busca do indicador URL, o processo de elaboração do *token* foi mais complexo, porque apesar dos URLs obedecerem aos standards definidos no RFC1630 têm muitas variantes que tornam o processo de definição *token* e consequente pesquisa extremamente complexo [16].

Para a definição deste *token* foi criado um conjunto de regras às quais o *token* tem de obedecer:

1. O URL deve começar por um dos seguintes conjuntos de caracteres:
   - http
   - https
   - hxxp
   - hxxps
2. Cada uma das hipóteses definidas anteriormente pode ser sucedida com a *string* 'www.'.
3. O URL pode conter os seguintes caracteres não alfanuméricos:
   - ?
   - =
   - &
   - %
   - .
   - –
   - +
   - [
   -]
   - /
4. Os grupos de caracteres alfanuméricos podem ser intercalados com um ou mais caracteres estipulados no grupo anterior de caracteres não alfanuméricos.
5. Os grupos de caracteres alfanuméricos têm um limite mínimo de três caracteres por grupo.
6. Os grupos de caracteres alfanuméricos não têm um limite máximo.

O *token* da expressão regular que definimos de acordo com os requisitos estipulados anteriormente está representado na Figura 4.4.

```
((hxxp:\/\/|hxxps:\/\/|http:\/\/|https:\/\/)?(\w*\D(\.|\[\.\]|\/|\-
|\?|\+|\=|\%|\&))+\w{1,})
```

Figura 4.4 - Token utilizado na busca por domínios.

4.2.3 Token para recolha de hashes

Para definir o *token* que efectua pesquisas do IOC hash tivemos em conta as principais funções de hash existentes, tais como md5, SHA-1 ou SHA-256, e os

respectivos resultados que cada uma produz: 128-bit, 256-bit e 512-bit, ou seja 32, 40 e 64 caracteres respectivamente.

Tendo em conta que as funções de hash mais utilizadas produzem resultados que contêm entre 31 e 65 caracteres, definimos que o indicador de compromisso hash deve ser composto por caracteres alfanuméricos em conjuntos de 32, 40 e 64 caracteres. Este *token* retorna resultados correspondentes aos algoritmos md5, SHA-1 ou SHA-256. O *token* definido está representado na Figura 4.5.

```
[a-f0-9]{128}((?=\ |\r|\n)|(?!\w|\d)))|([a-f0-9]{64}((?=\
|\r|\n)|(?!\w|\d)))|([a-f0-9]{40}((?=\ |\r|\n)|(?!\w|\d)))|([a-f0-9]{32}((?=\
|\r|\n)|(?!\w|\d)))
```

Figura 4.5 - Token utilizado na busca por hash.

4.3 Validação dos indicadores recolhidos

Neste subcapítulo iremos abordar as diferentes formas utilizadas na validação dos metadados recolhidos sobre cada indicador. Iremos também explicar quais os parâmetros de avaliação definidos para os diferentes indicadores de acordo com os critérios aplicados no SOC do MAI.

O processo de validação de domínios/URLs e dos hashes só é passível ser efectuado recorrendo a aplicação externa, sendo que neste projecto recorremos ao VirusTotal, disponível online no site *www.virustotal.com*. Foi escolhida esta aplicação porque é uma das ferramentas com maior fiabilidade, que disponibiliza uma API gratuita que pode ser usada por qualquer utilizador para fazer pedidos Hypertext Transfer Protocol (HTTP). O VirusTotal permite analisar indicadores de compromisso através da consulta de bases de dados de diversas empresas que produzem software, como por exemplo antivírus, bem como a consulta de uma BD de relatórios feitos por utilizadores registados na plataforma. Desta forma é possível obter uma informação relativamente fiável quando se trata de IOCs conhecidos ou com uma elevada taxa de disseminação em plataformas semelhantes online.

Através da API do VirusTotal é possível efectuar enúmeras operações, sendo que no contexto deste projecto focámo-nos em:

- Submeter um indicador para análise (URL e hash);
- Obter o relatório da análise do indicador.

Para efectuar os pedidos HTTP às APIs das aplicações utilizamos a biblioteca *requests.py* porque esta permite configurar o cabeçalho do pedido HTTP e fazer pedidos autenticados. Para efectuar pedidos autenticados às aplicações Apility e VirusTotal foi necessário efectuar um registo em cada uma delas para incluir o *token* de autenticação no cabeçalho de cada um dos pedidos.

4.3.1 Validação dos IPs

No processo de recolha dos IPs é evitar o armazenamento de IPs que não correspondem aos standards existentes do IPv4 [14]. O processo de validação está descrito na Figura 4.6.

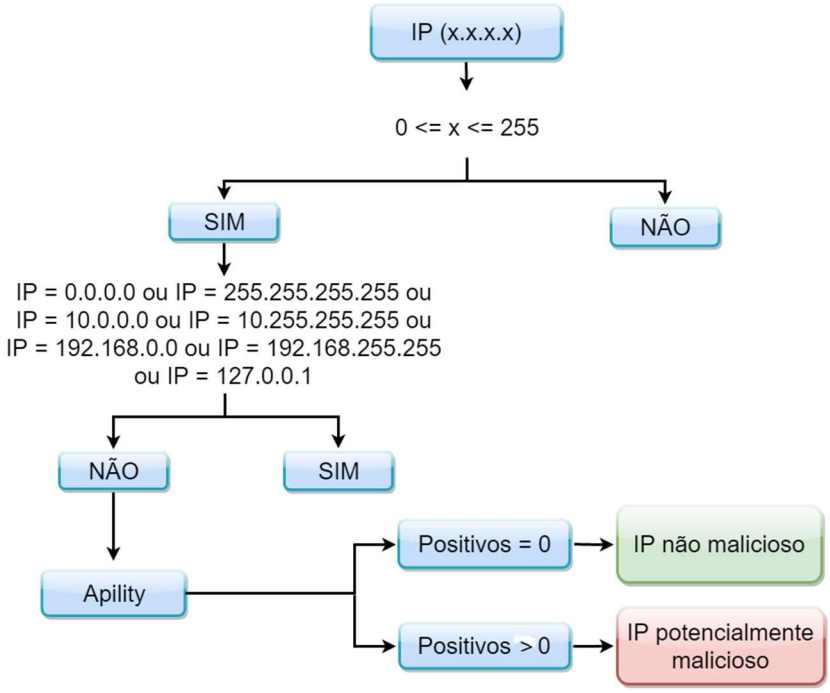

Figura 4.6 - Validação de IPs.

Os IOCs relativos a IPs que estão armazenados na colecção 'ip' da BD osintdb são armazenados pelo módulo indicators.py com os seguintes metadados: ip, data, país, resposta do Apility, pontuação, URL da notícia e etiquetas.

Do ponto de vista de análise OSINT é necessário obter informação relativa aos indicadores recolhidos para efectuar validações que reduzam a margem de erro. A recolha deste tipo de informação permite remover IPs que foram recolhidos das notícias e que não apresentam indícios maliciosos, associar tipos específicos de ataques a um determinado IP ou até associar ao IP um indicador de nível de confiança.

Para efectuar a recolha deste tipo de informações existem diversas aplicações web que disponibilizam, de forma gratuita, APIs que permitem automatizar o processo de recolha de metadados relativos a um recurso específico. Para atingir este objectivo foram criados diversos procedimentos de recolha e validação de dados em cada um dos métodos executados pelo módulo indicators.py. As informações mais específicas sobre um IP, no contexto de um SOC, são:

- Nível de confiança do IP em análise
- Tipos de ataques em que este IP esteve envolvido
- Internet Service Provider (ISP) e país
- Domínios associados a este IP

A recolha deste tipo de informação sobre um IP tem 2 grandes objectivos: 1) fornecer dados que permitam contextualizar o analista para que possa ser efectuada uma análise mais completa e rigorosa; 2) recolha de metadados para um processo de registo mais completo no MISP. A informação que é colocada no MISP, por exemplo através de etiquetas, irá permitir um melhor correlacionamento de informação, como iremos analisar no subcapítulo da "Integração com MISP". Este tipo de informação permite que o analista consiga correlacionar o IP em causa de uma forma mais rápida e eficiente.

4.3.2 Validação de domínios/URLs

Tendo em conta a hierarquia representada num URL, não é possível determinar se o URL em causa representa todos os níveis desta hierarquia. Por isso, para validarmos se um URL é malicioso recorremos ao VirusTotal e analisamos o número de resultados positivos, que representa a quantidade de BDs em que este URL foi encontrado. Caso o URL seja não malicioso ou inválido a resposta de VirusTotal irá devolver um número de resultados positivos igual ao zero. Se o URL em causa for malicioso ele estará presente nas bases de dados consultados pelo VirusTotal, devolvendo como resultados positivos o número de bases de dados em que o URL está presente. Desta forma conseguimos depreender a partir da resposta do VirusTotal se o URL é válido e malicioso; válido e não malicioso; ou inválido. O processo de validação dos domínios recorre também à API do VirusTotal para efectuar os mesmos procedimentos de validação. O processo de validação de um domínio/URL está demonstrado na Figura 4.7.

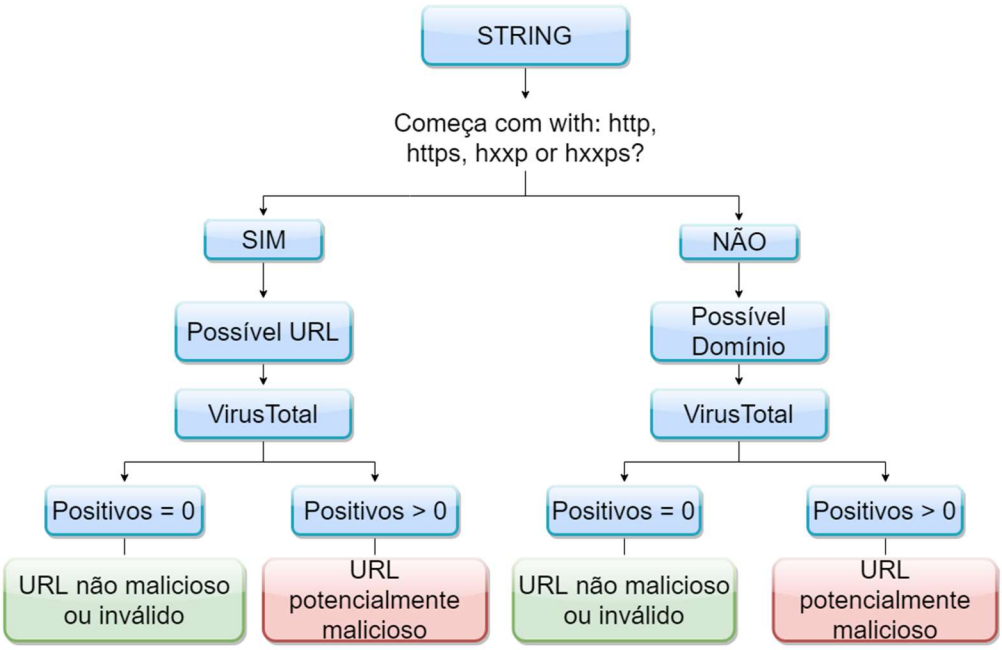

Figura 4.7 - Validação domínios/URL.

4.3.3 Validação de hash

O processo de validação do IOC hash é um conjunto de procedimentos semelhante ao processo de validação de domínios /URL, na medida em que existe um conjunto de caracteres que passa por um processo de validação inicial e é submetido à posteriori para validação do mesmo. Tal como foi explicado anteriormente, os IOCs do tipo hash que são encontrados nas notícias podem ter 30, 40 ou 64 caracteres. Estes conjuntos de caracteres são submetidos a análise na aplicação VirusTotal e o resultado dessa análise permitirá saber: qual o tipo de hash, se é válido ou não; e se é ou não malicioso. Este processo está descrito na Figura 4.8.

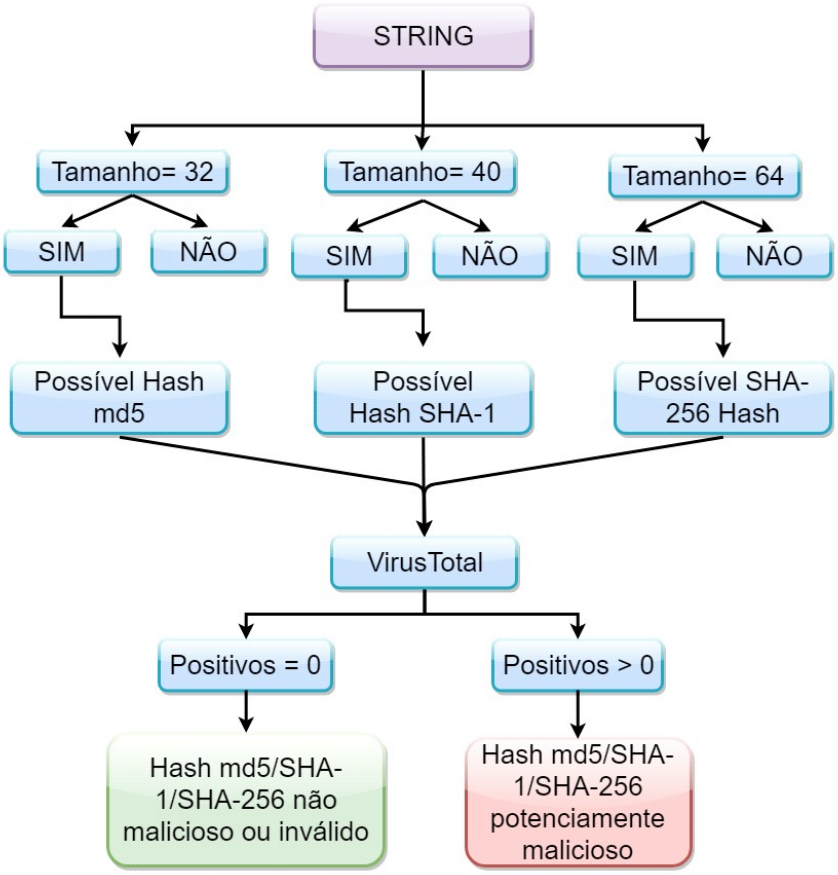

Figura 4.8 - Validação de hash.

4.4 Armazenamento de dados

MongoDB é uma BD não relacional que permite armazenar grandes quantidades de dados. O armazenamento dos dados é feito num esquema semelhante ao JavaScript Object Notation (JSON). O facto de armazenar os dados num esquema semelhante ao JSON e de permitir exportar os dados de uma colecção em *arrays* de objectos JSON, foram algumas das razões para a nossa implementação de uma base de dados MongoDB.

Os dados armazenados na BD são na sua maioria sequências de caracteres ou texto, *strings*, ou estruturas de dados que são resultado do processamento da informação, como por exemplo *arrays*. A grande maioria dos dados armazenados são notícias recolhidas pelo módulo zavot.py.

Existem 5 colecções de dados na base de dados OSINTDB:

- news
- keywords
- hash
- domain
- ip

A escolha para o armazenamento dos dados nestas colecções teve em conta o facto de ser necessário a separação de IOCs para pós-processamento dos mesmos através do módulo de Python indicators.py.

Neste projecto a BD OSINTDB é criada localmente durante a configuração inicial da máquina, sendo à posteriori actualizada em cada iteração do módulo.

4.4.1 Armazenamento de notícias

A colecção news agrega todos os objectos que representam as notícias recolhidas a partir dos *feeds*.

O módulo zavot.py recolhe e processa as notícias que são obtidas nos *feeds* dos diferentes sites em cada interacção. De seguida o módulo zavot.py envia os dados de cada notícia ao módulo mongo.py, para que este possa validar se a notícia existe ou não na BD.

Para que não existissem problemas em termos de armazenamento ou eficiência na pesquisa de indicadores, procurámos com este modelo de dados armazenar os dados estritamente necessários ao funcionamento da Zwerg, evitando assim inserir informação desnecessária na BD que iria tornar lenta a recolha, o armazenamento e a pesquisa de informação numa BD que se pretende manter escalável.

A estrutura de dados de cada objecto que compõe a colecção news é semelhante a um JSON pares chave valor. Cada um destes campos é recolhido a partir do elemento item que esta presente no *feed* de notícias. Os campos escolhidos para compor cada um dos objectos são: title, url, pubDate, description e content, como demonstra a Figura 4.9.

```
{
    title : 'Título da notícia',
    url : 'url da notícia',
    pudDate : 'data de publicação da notícia',
    description : 'descrição da notícia',
    content : 'conteúdo da notícia'
}
```

Figura 4.9 - Estrutura de dados da colecção news.

O título e a descrição da notícia, inseridos nos campos title e description respectivamente, são essenciais para que o analista possa contextualizar os IOCs recolhidos de uma forma mais precisa. Por exemplo, na busca de informações sobre um determinado URL ou IP pode ser necessário consultar a notícia por forma a perceber qual é o contexto em que os indicadores estão inseridos.

O URL da notícia é o elemento utilizado para distinguir entre as diferentes notícias que são inseridas na colecção. Quando o módulo zavot.py recolhe as notícias envia-as para o mongo.py para que estas sejam inseridas na colecção. No entanto, como os módulos são executados ciclicamente, a notícia pode já existir na BD. Para evitar a duplicação de entradas na BD o módulo mongo.py utiliza o URL de cada notícia como identificador único. Se a notícia já existir na BD utilizámos a data de publicação, pudDate, para actualizar a notícia existente na BD de forma a guardar apenas o conteúdo mais recente.

O conteúdo da notícia é armazenado, em cada objecto, associado no campo content.

4.4.2 Armazenamento de IPs

A colecção IP armazena todos os indicadores IP que são recolhidos a partir dos *feeds* de notícias. A estrutura de dados de cada objecto IP é composta por 7 campos: ip, date, country, abuse, score, url e tags.

O campo IP regista o endereço IP, o indicador que foi recolhido da notícia. O campo date regista a data em que o objecto foi criado ou actualizado. O country é um código de duas letras que representa o país em que o IP se encontra registado. O campo abuse é uma estrutura de dados em formato JSON que contém informações recolhidas a partir da aplicação *Apility*. O resultado de uma pesquisa à aplicação é um conjunto de dados com diversas informações, tais como a geolocalização, nível de abuso do IP, *blacklists* em que o IP se encontra registado, entre outros. Apesar de apenas ser considerado o nível de abuso do IP e as bases de dados em que se encontra registado, decidimos guardar na BD toda estrutura da resposta proveniente da aplicação Apility pois ao manter a mesma estrutura torna-se mais fácil o desenvolvimento de novas funcionalidades na aplicação, a partir de dados presentes nesta estrutura, porque a estrutura de dados se manteve fiel à resposta obtida no pedido original. O campo score contém a pontuação atribuída pelo Apility ao IP em questão. O Campo url contém o URL da notícia onde o IP foi encontrado. O

campo tags contém uma lista de etiquetas que foram encontradas na notícia onde o IP foi encontrado. O esquema de armazenamento de IPs está ilustrado na Figura 4.10.

```
{
    ip : IOC em análise,
    date : data no formato 'AAAA-MM-DD HH:MM',
    country : código ISSO, de 2 caracteres,
    abuse : { resposta JSON da aplicação Apility},
    score : score do ip,
    url : url da notícia,
    tags : [tag1, tag2, tag3]
}
```

Figura 4.10 - Estrutura de dados da colecção IP.

4.4.3 Armazenamento de domínios/URLs

A colecção domain contém objectos que agregam toda a informação relacionada com o indicador domínio/URL. A estrutura base contém 7 campos: url, date, vtDomain, *link*, vtDomainFlag, orderFlag e tags. Esta estrutura pode ser visualizada na Figura 4.11.

```
{
    url : IOC em análise,
    date : data no formato 'AAAA-MM-DD HH:MM',
    vtDomain : {
        url : www.example.com,
        response_code : 1,
        scans : {
            database1 : {
                detectec : true,
                detail : url do resource on database 1,
                result : suspicious site
            },
            ...
        },
        filescan_id : id do scan do ficheiro, se existir,
        positives : número de positivos nas bases de dados,
        scan_date : AAAA-MM-DD HH:MM:SS,
        resource : www.example.com,
        verbose_msg : scan finished, information embedded in this object,
        scan__id : id do scan,
        permalink : url da página do scan,
        total : número total de bases de dados consultadas
    },
    link :'url da notícia',
    vtDomainFlag : 'CRITICAL',
    orderFlag : pontuação do domínio,
    tags : [tag1, tag2, tag3]
}
```

Figura 4.11 - Estrutura de dados da colecção domain.

O campo url contém o domínio/URL que foi encontrado na notícia e submetido para análise. O campo date é utilizado para guardar a data em que foi efectuado registo na BD.

No processo de análise de um determinado indicador URI efectuamos um pedido à aplicação VirusTotal. A resposta resultante desse tem um formato JSON e é guardada no campo vtDomain. Mais uma vez armazenamos toda a resposta, ao invés de campos específicos, pelas mesmas razões que armazenamos toda a resposta proveniente do pedido de análise ao indicador IP. O campo link contém o *link* da notícia onde o IOC em questão foi encontrado. O campo vtDomainFlag contém a classificação que foi associada ao IOC, em função do número de positivos da análise do VirusTotal. O campo orderFlag contém um valor, que varia entre 0 e 3 correspondente ao campo vtDomainFlag, sendo que 0 significa SAFE e 3 significa CRITICAL. O campo tags contem uma lista de 3 *strings* em que cada uma das *strings* corresponde a uma etiqueta que foi encontrada na notícia.

4.4.4 Armazenamento de hashes

A colecção hash contém objectos representativos do hash e respectiva análise obtida através da API do VirusTotal. Os campos hash e date representam a hash e a data em que foi inserida na BD respectivamente. Cada hash pode ter 30, 40, 128 e 256 caracteres. O campo vtDomain representa a resposta, em formato JSON, do pedido de análise feito à API do VirusTotal. O campo link contém o *link* da notícia onde o hash foi encontrado. O campo vtHashFlag contém a pontuação que foi associada ao hash, em função do número de positivos da análise do VirusTotal. O campo orderFlag é semelhante ao campo com o mesmo nome no objecto url da colecção domains. O campo tags contém uma lista de 3 *strings* em que cada uma das *strings* corresponde a uma tag que foi encontrada na notícia. A estrutura de dados é apresentada na Figura 4.12.

```
{
    hash : 'a8f5f167f44f4964e6c998dee827110c' ,
    date : data no formato 'AAAA-MM-DD HH:MM',
    positives : número de positivos na análise do VirusTotal,
    vtAnalysis : {
        url : www.example.com,
        response_code : 1,
        scans : {
            database1 : {
                detected : true,
                detail : url da BD,
                result : suspicious site
            },
            …
        },
        filescan_id : id do scan do ficheiro, se existir,
        positives : número de positivos nas bases de dados,
        scanDate : AAAA-MM-DD HH:MM:SS,
        resource : www.example.com,
        verbose_msg : scan finished, information embedded in this object,
        scanId : id do scan,
        permalink : url da página do scan,
        total : número total de bases de dados consultadas
    },
    link : URL da página do scan,
    vtHashFlag : WARNING,
    tags : [tag1, tag2, tag3]
}
```

Figura 4.12 - Estrutura de dados da colecção hash.

4.5 Integração com o MISP

O Malware Information Sharing Platform (MISP) é um software de armazenamento e partilha de *intelligence* que permite armazenar grandes quantidades de informação de forma organizada. O MISP é frequentemente utilizado na área da segurança informática para o armazenamento de IOCs por forma a ter um ponto comum de agregação de informação. Existem duas grandes vantagens na utilização do MISP como ferramenta de armazenamento de informação: a agregação de dados e consequente correlação que é gerada, através de visualização de dados, entre os diferentes eventos que contêm atributos em comum; a vasta panóplia de integrações, que existe em diversas linguagens de programação, com diversas aplicações e software para recolha de da dos de fontes externas, como por exemplo o OTX , SPLUNK, QRADAR, entre outros. A integração entre a Zwerg e o MISP é feita em Python 3 com recurso à biblioteca pymisp.py.

O MISP é uma ferramenta que funciona com base em eventos. Estes eventos guardam diversas informações sobre um determinado IOC, ataque, ou registo de incidente. O processo de produção destes eventos é constituído por três fases:

1. Criar o evento na plataforma MISP
2. Popular o evento com diversos atributos e metadados para enriquecer o IOC
3. Publicar o evento.

A Zwerg recolhe os três principais tipos de indicadores: IPs, domínios/URLs e hashes. Como foi explicado anteriormente os indicadores são armazenados em bases de dados separadas, por forma a manter organizada a informação nos vários estados ao longo do processo: recolha, normalização, processamento, armazenamento e inserção no MISP. Tendo este aspecto em mente considerámos que faz sentido separar os vários tipos de indicadores em vários eventos. Por isso, são criados três eventos diferentes por dia no MISP, sendo que cada um deles corresponde a um tipo de IOC. A distinção entre os vários eventos que são criados diariamente é feita em dois aspectos: a info, que contém o tipo de indicadores armazenados nesse evento e tipo que é associado a cada atributo. Este último será abordado e detalhado nos capítulos em diante.

A criação de eventos e inserção de todos os indicadores na base de dados do MISP é da responsabilidade do módulo importToMisp.py. Quando um evento é criado contém apenas a informação genérica a todos os eventos: distribution, threat_level, analysis, info, date e organization. O parâmetro distribution é preenchido com o valor 0, que significa que os eventos publicados no MISP apenas serão visíveis dentro da organização. O parâmetro threath_level é preenchido a 1 que corresponde ao nível high.

Para evitar popular a BD do MISP com indicadores que geram falsos positivos escolhemos associar ao evento IOCs que obtiveram o nível de classificação diferente de SAFE. Ou seja, apenas os IOCs classificados com CRITICAL, WARNING ou SUSPICIOUS são inseridos no MISP. Os restantes IOCs não são inseridos no MISP. A opção de apenas inserir os IOCs com o nível de classificação CRITICAL tem a ver com o facto de a BD de eventos do MISP ser utilizada pelo SIEM para correlacionar

eventos com os *logs* em tempo real. Se todos os IOCs fossem inseridos no MISP a taxa de falsos positivos iria aumentar visto que nem todos os IOCs recolhidos pela Zwerg são necessariamente maliciosos. É por isso que a Zwerg efectua uma análise, através das várias aplicações enumeradas no subcapítulo 4.2 Processamento de Informação. O parâmetro analysis é preenchido com o valor 2 que corresponde a uma análise completa. Nesta fase já foram realizadas todas as análises e validações passíveis de serem realizadas através dos processos autónomos da Zwerg. O parâmetro info é preenchido com uma taxonomia própria, criada pelo MAI, que contém os seguintes dados: "CSIRT MAI – IOC from Infosec – Data da criação do evento", em que apenas o último componente varia em função do dia em que o evento é criado. O parâmetro date é preenchido com a data da criação do IOC e o parâmetro *organization* é preenchido com o nome da organização responsável pelos eventos do MISP. O resultado do processo de criação de eventos no MISP pode ser observado na Figura 4.13.

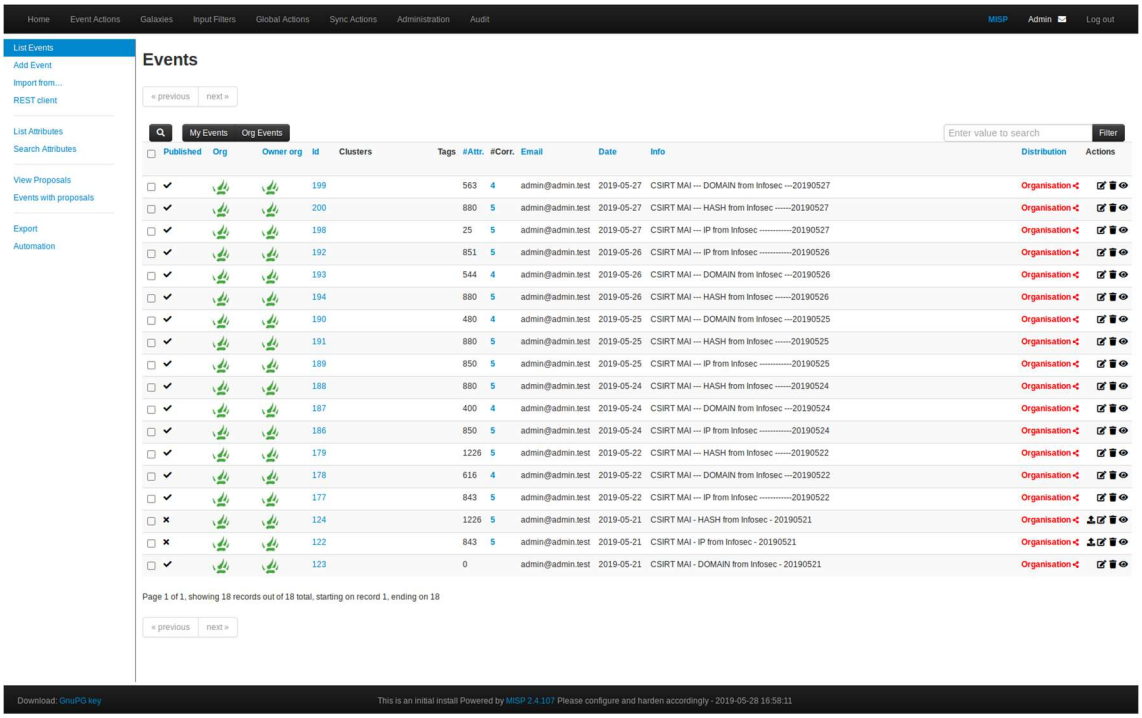

Figura 4.13 - Eventos do MISP

Como é possível perceber nesta fase os eventos ainda não contêm os indicadores de compromisso associados pelo que é necessário associar os vários tipos indicadores ao respectivo evento. Para isso são recolhidos, a partir das colecções do MongoDB os indicadores que devem popular cada um dos eventos criados MISP.

Cada IOC armazenado na base de dados MongoDB tem associada uma lista de etiquetas. Esta lista de etiquetas é composta por três palavras ou expressões, encontradas na notícia de onde o IOC foi recolhido, que melhor descrevem o indicador de compromisso. Esta lista pode ser consultada no Anexo C. A título de exemplo o URL hxxp://malicious_domain.example contém a seguinte lista de etiquetas associadas a si: "Minning", "Malware" e "Encryption". Estas tags podem ser

usadas com os seguintes objectivos: elucidar o analista acerca de possíveis ataques ou objectivos de um determinado indicador, como também gerar correlações no MISP entre os vários atributos de um ou mais eventos. Estas correlações podem ser úteis para perceber qual a tendência de ataques dos últimos indicadores recolhidos pela Zwerg. Para aumentar a objectividade das etiquetas que são associadas a cada um dos atributos do MISP cada indicador contém no máximo 3 etiquetas.

A título de exemplo demonstramos através da tabela 4.1 uma estrutura simplificada de um evento do MISP

No dia 1 de janeiro de 2019 foi recolhida a seguinte lista de indicadores:

- ip1.ip1.ip1.ip1 - [‘tag 1’, ‘tag 2’, ‘tag 3’]
- domain1.example.com - [‘tag 4’, ‘tag 5’, ‘tag 6’]
- ip2.ip2.ip2.ip2 - [‘tag 7’, ‘tag 8’, ‘tag 9’]
- ip3.ip3.ip3.ip3 - [‘tag 10’, ‘tag 11’, ‘tag 12’]
- ip4.ip4.ip4.ip4 - [‘tag 13’, ‘tag 14’, ‘tag 15’]
- …
- domain2.example.com - [‘tag 16’, ‘tag 17’, ‘tag 18’]
- aafd323423422aca34234ba3s2345eb - [‘tag 19’, ‘tag 20’, ‘tag 21’]
- 76874c4a45s6f5c54a54de66f5c54a24 - [‘tag 22’, ‘tag 23’, ‘tag 24’]

O evento criado neste dia tem um formato que pode ser observado na Tabela 4.1.

Evento criado no MISP pela Zwerg		
Title	ORG – IOC from Infosec – 20190101	
Date	01-01-2019	
Distribution	Your organization only	
Threat Level	High	
Event Description	IOCs collected from Intel gathering on 01-01-2019	
Tags	[‘tag 3’,’tag 8’,’tag 9’]	
Atributes	IOC	Tags
	ip1.ip1.ip1.ip1	[‘tag 1’, ‘tag 2’, ‘tag 3’]
	domain1.example.com	[‘tag 4’, ‘tag 5’, ‘tag 6’]
	ip2.ip2.ip2.ip2	[‘tag 7’, ‘tag 8’, ‘tag 9’]
	ip3.ip3.ip3.ip3	[‘tag 8’, ‘tag 11’, ‘tag 12’]
	ip4.ip4.ip4.ip4	[‘tag 13’, ‘tag 14’, ‘tag 9’]
	…	…
	domain2.example.com	[‘tag 16’, ‘tag 7’, ‘tag 3’]
	aafd323423422aca34234ba3s2345eb	[‘tag 9’, ‘tag 20’, ‘tag 21’]
	76874c4a45s6f5c54a54de66f5c54a24	[‘tag 22’, ‘tag 3’, ‘tag 9’]

Tabela 4.1 - Estrutura de dados do evento do MISP.

Após serem preenchidos com atributos todos os eventos criados no dia, são associadas 3 etiquetas a cada um dos eventos. O principal objectivo destas etiquetas

é salientar quais os principais destaques associados aos indicadores armazenados no MISP durante um período de seis meses prévios à criação do evento. Ou seja, a evolução das etiquetas permite avaliar a evolução da cibersegurança nos últimos 6 meses, em função dos principais destaques das notícias.

O resultado da inserção dos atributos nos eventos do MISP pode ser visualizado na Figura 4.14.

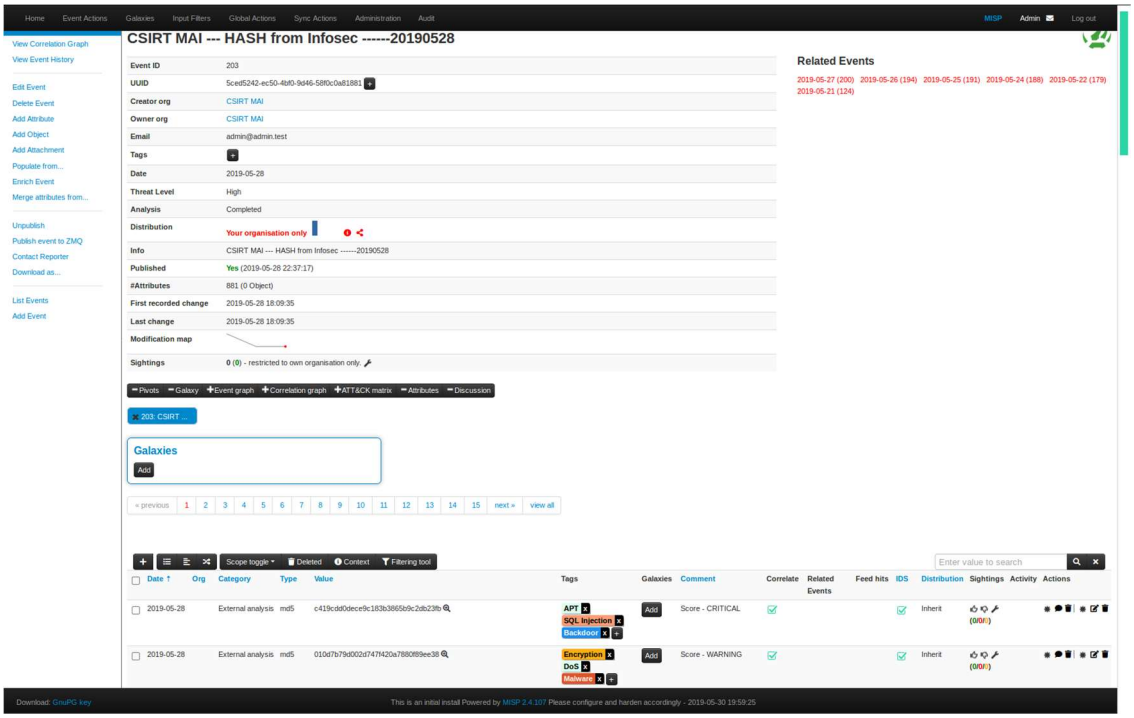

Figura 4.14 - Atributos do Evento do MISP.

Capítulo 5 **Aplicação Web**

Neste capítulo iremos explicar o modo de funcionamento da aplicação web que criámos para a Zwerg e quais os seus principais objectivos. Este capítulo está dividido em 4 subcapítulos que abordam os principais objectivos da aplicação e as razões que levaram à criação de cada uma das funcionalidades. Neste capítulo serão abordadas apenas as principais funcionalidades do ponto de vista de utilização prática no dia a dia de um SOC. Contudo, existem outras funcionalidades na aplicação que poderão ser utilizadas menos frequentemente, mas com igual importância, tais como os gráficos ou a busca nas notícias por palavras específicas.

5.1 Objectivos

A criação desta aplicação deveu-se principalmente à necessidade ter de uma resposta rápida a incidentes de segurança informática no SOC. As principais razões que motivaram a criação desta aplicação foram: facilitar o acesso aos diversos tipos de dados existentes na BD para uma rápida interacção com os mesmos; tornar a visualização dos dados mais apelativa e eficiente, por exemplo através de tabelas que agregam os dados; permitir a consulta e edição de configurações necessárias ao funcionamento da Zwerg; criação de gráficos para visualização de metadados; usabilidade necessária no processo de configuração da Zwerg.

A criação da aplicação web não era um dos objectivos inicialmente alinhavados pelo COSI e partiu de uma necessidade encontrada durante o processo de testes. As questões de usabilidade e visualização da informação levaram a que fosse desenvolvida esta interface web de modo a tornar mais fáceis estes processos.

5.2 Visualização de dados

A Zwerg armazena toda a informação em bases de dados que podem ser consultadas pelos utilizadores. No entanto a consulta a uma BD implica ter conhecimento dos comandos de interacção com a BD. Por outro lado, a consulta de informação relativa a um IOC, numa base de dados MongoDB, pode tornar-se complexa quando é necessário efectuar uma pesquisa rápida e objectiva, no que toca aos diferentes resultados a serem apresentados.

Numa pesquisa efectuada à sobre um IP são mais relevantes os dados que informam as principais características desse IOC. No caso do IP esses dados são: o país, a sua criticidade definida com base no relatório da aplicação web Apility e a data em que foi inserido na BD.

Por forma a facilitar o acesso e manuseamento dos dados por parte do utilizador final, concebemos um conjunto de páginas na aplicação web que permitem a interacção e visualização dos dados. Na página principal definimos três tabelas,

sendo cada uma delas relativa a cada um dos IOCs recolhidos pela Zwerg, respectivamente IP, URL e HASH.

A tabela IP, apresentada na página principal da ZEWRG, contém 4 colunas: IP, COUNTRY, DATE e CRITICALITY. Cada uma destas colunas permite ao analista obter informação essencial sobre o IOC em análise. Por exemplo a coluna COUTRY permite saber qual o país relativo ao IP. Países com a China, Rússia ou Irão são conhecidos nos SOCs por serem frequentemente países associados a diversos tipos de ataques informáticos de grande escala e complexidade. A coluna DATE permite que o analista perceba se o IP foi inserido na BD num período recente ou se se trata de informação em fim de prazo que em breve será removida da aplicação. A coluna CRITICALITY representa o qual a pontuação, atribuída pelo COSI, a este IP com base na resposta obtida no relatório da aplicação Apility. A escala de valores foi definida no SOC do MAI e divide-se em quatro: SAFE, SUSPICIOUS, WARNING e CRITICAL.

A aplicação Apility tem um standart de reposta que varia entre -3 e 0. A conversão desta pontuação para a escala de pontuação definida pelo SOC do MAI é apresentada na Tabela 5.1.

Pontuação Apility	Escala da Zwerg
0	SAFE
-1	SUSPICIOUS
-2	WARNING
-3	CRITICAL

Tabela 5.1 - Relação entre a pontuação do Apility e a classificação atribuída pelo MAI.

A tabela dos URL contém 3 colunas: URL, DATE e CRITICALITY. Tal como na tabela dos IP, cada coluna contém informação relativa ao IOC. A coluna DATE representa a data em que o URL foi inserido na BD ao passo que a coluna CRITICALITY representa a pontuação, convertida para a escala definida pelo SOC do MAI, obtida na resposta dada pela aplicação VirusTotal. Os valores atribuídos pela aplicação VirusTotal representam o número de bases de dados em que o IOC foi encontrado. A relação entre a escala definida pelo COSI e o número de positivos é demonstrada na Tabela 4.2.

Número de positivos VirusTotal	Escala da Zwerg
0	SAFE
1	SUSPICIOUS
2	WARNING
>=3	CRITICAL

Tabela 5.2 - Relação entre a pontuação do VirusTotal para domínios/URLs e classificação atribuída pelo MAI.

A tabela HASH contém 3 colunas: HASH, DATE e CRITICALITY. À semelhança das tabelas anteriores as diferentes colunas apresentam os principais dados sobre cada um dos indicadores encontrados. A conversão entre a o número de positivos provenientes da resposta do VirusTotal e a escala do SOC do MAI é apresentada na Tabela 5.3.

Número de positivos no VirusTotal	Escala da Zwerg
0	SAFE
0 < Positivos <= 4	SUSPICIOUS
4 < Positivos <= 8	WARNING
Positivos >8	CRITICAL

Tabela 5.3 - Relação entre a pontuação do VirusTotal para hashes e classificação atribuída pelo MAI.

O resultado deste processo pode ser visto na Figura 5.1

Figura 5.1 - Página Principal.

5.3 Consulta de informação detalhada

A informação apresentada na página principal da Zwerg é essencial para que o analista possa, de uma forma rápida, perceber se o IOC em questão representa uma ameaça aos sistemas informáticos do MAI. É possível depreender esta informação nos casos em que o IOC tenho associado uma pontuação de CRITICAL. No entanto esta informação podes ser, por vezes, insuficiente para que se possa estabelecer uma ideia clara acerca do IOC em questão. Por isso, e com o intuito de tornar a análise do analista mais completa, decidimos disponibilizar toda a informação proveniente das respostas dadas pelas aplicações web Apility e VirusTotal.

Por forma a disponibilizar toda a informação relativa a um IOC, obtida durante o processo de recolha de indicadores, foram criadas 3 páginas: IP, HASH e URL. Cada uma destas páginas contém um formulário que permite a pesquisa de informação relativa a um determinado IOC. No caso dos IPs é apenas necessário inserir o IP em análise e, caso ele se encontre na BD, será disponibilizada toda a informação através

de várias tabelas que agregam os dados os diferentes conjuntos de dados. Estes conjuntos de dados vão desde informações variadas, como por exemplo país de registo do IP, a data em que foi realizada a última análise ao IOC ou o conjunto de bases de dados em que o IP se encontra registado como malicioso. As páginas HASH e URL cumprem o mesmo objectivo, sendo que apresentam também informação complementar sobre as bases de dados em que o IOC foi encontrado bem como as etiquetas a ele associadas.

A consulta de informação detalhada, precisa e organizada permite que o analista seja capaz de efectuar uma análise mais completa e eficaz. A Figura 5.2 e a Figura 5.3 demonstram o processo de pesquisa de informação detalhada sobre um hash.

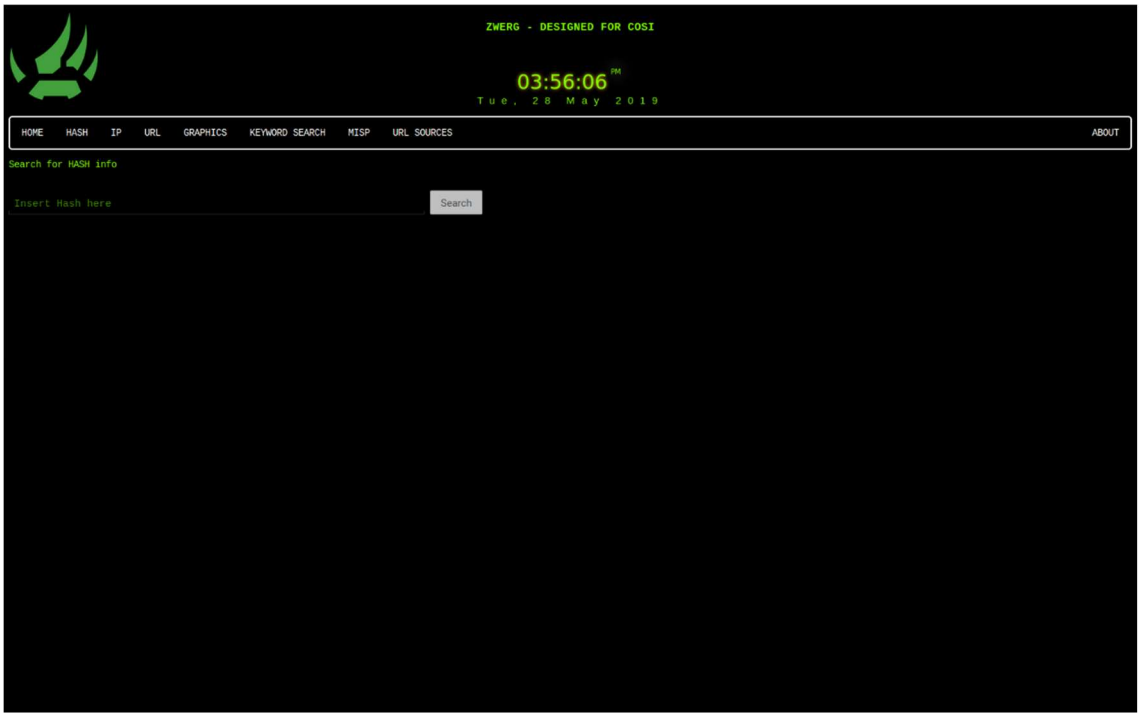

Figura 5.2 - Página de pesquisa de hash.

Figura 5.3 - Resultados da pesquisa por hash.

5.4 Configuração dos feeds

A Zwerg permite obter dados de diversos *feeds* de informação. A quantidade de informação recolhida depende do número de *feeds* configurados na Zwerg. A evolução tecnológica nas diversas áreas afectas à segurança informática leva a que exista cada vez mais conteúdo de relevo produzido por diversos autores com elevada importância no contexto de um SOC. A distribuição deste conteúdo é feita maioritariamente através de *blogs* de investigadores e especialistas em segurança informática. Cada vez mais existem *blogs* e plataformas de notícias online que publicam conteúdo essencial aos analistas de um SOC.

A análise de informação efectuada pela Zwerg depende dos *feeds* que são dados como input inicial para recolha de informação através dos *feeds* distribuídos pelas plataformas de distribuição de conteúdo online. Por isso surgiu a necessidade de tornar o processo de configuração dos URLs dos *feeds* acessível ao utilizador final.

Os *feeds* são ficheiros XML que podem ser acedidos através de um endereço URL. Com a constante adição de conteúdo na Internet surgiram também novas plataformas e *blogs* de analistas. Estas novas plataformas contêm novos endereços de *feeds* que podem ser adicionados à lista inicial de endereços de *feed* de recolha de informação. Por esta razão surgiu a necessidade de facilitar o acesso a esta configuração através da página “URL Sources”.

A lista de *feeds* utilizada no processo de desenvolvimento e testes da Zwerg pode ser consultada no Anexo B. A Figura 5.4 demonstra a página de configuração de URLs de *feeds*.

Figura 5.4 - Configuração de URLs de feeds.

Capítulo 6 **Resultados**

Neste capítulo iremos abordar o processo de validação dos resultados obtidos pelos vários módulos da Zwerg. Este processo consiste na apreciação efectuada pelo Manager do SOC do MAI aos resultados produzidos pela ZEWRG bem como a nossa avaliação realizada à eficácia e eficiência dos diversos componentes desenvolvidos durante o processo de elaboração deste projecto.

6.1 Avaliação dos Módulos

A Zwerg é constituída por diferentes módulos, sendo que cada um deles tem um objectivo único e preciso. Por forma a efectuar uma validação de resultados justa, é necessário validar o resultado produzido por cada um dos módulos de forma independente. Para cada um dos módulos foi elaborada uma lista de testes que permite aferir o sucesso de cada dos procedimentos definidos no módulo.

Lista dos testes a serem efectuados em cada módulo:

- Zavot
  - Sucesso do armazenamento das notícias para análise
  - Erros obtidos no processo de recolha de notícias
- Indicators
  - Etiquetas recolhidas para cada indicador
  - Percentagem de eficácia na análise dos indicadores (está dependente de aplicações externas)
- Import to MISP
  - Validar a criação de eventos de acordo com a estrutura descrita no capítulo Integração com o MISP
  - Validar o processo de inserção de atributos nos eventos
  - Etiquetas associadas a cada evento do MISP

A partir dos testes efectuados ao módulo Zavot foi possível aferir, através de diversas execuções do software, que a média de notícias recolhidas por iteração é de cerca de 3000 notícias. Das 3000 notícias recolhidas, cerca de 2100 são recolhidas com sucesso. As cerca de 900 que não são recolhidas com sucesso não são armazenadas na BD. Isto deve-se ao facto de o *feed* de onde a notícia é recolhida não ter todos os campos preenchidos. Com estes dados podemos concluir que 70% das notícias são recolhidas com sucesso, ou seja com todos os campos devidamente preenchidos.

O módulo Indicators procede à análise de cada um dos indicadores, presente na BD, individualmente, ou seja, para cada IOC é feito um pedido HTTP à aplicação externa que avalia o IOC. Os resultados destes pedidos, após executar o software, pelo menos 5 vezes por dia, durante o período de 3 semanas revelam que: foram efectuados diversos pedidos para cada um dos 644 domínio/URL e 627 desses pedidos, cerca de 97.4%, obtiveram resposta positiva enquanto 17 pedidos

continuam à espera de resposta; os pedidos executados nas iterações dos testes efectuados, para cada um dos 856 IPs existentes na BD obtiveram 601 respostas positivas, cerca de 70.2%; para cada um dos 1247 hashes existentes na BD foi obtida uma resposta com sucesso, o que representa uma eficácia de 100%. O processo de recolha de etiquetas para associar aos indicadores revelou-se extremamente eficiente, visto que todos os indicadores têm pelo uma etiqueta associada.

Para garantir a execução do requisito funcional de criação de eventos no MISP, foram realizados testes ao módulo "Import to MISP" cujo resultado pode ser visualizado na Figura 4.13, onde é possível verificar que o processo de criação de eventos e associação de um nível de criticidade é eficaz. Para confirmação dos resultados inseridos nos atributos foram efectuados pedidos à aplicação VirusTotal, através da interface web, que confirmaram a veracidade do nível de criticidade dos IOCs associados aos eventos.

Por forma a cumprir os requisitos de usabilidade foi desenvolvida uma aplicação interactiva, de fácil utilização e que usa um esquema de cores semelhante ao esquema utilizado pelas aplicações presentes no videowall. O resultado destes testes pode ser observado pelos Screenshots da aplicação no Anexo D.

Foram também realizados testes para a verificação de erros nas execuções iterativas do software. Durante o período de testes não foi encontrado qualquer erro. Este cenário deve-se ao facto de todos os possíveis cenários passíveis de originar erros serem tratados através de excepções. Importa dizer que este processo de validação de possíveis erros foi executado durante todo o período de desenvolvimento do software afecto a este projecto.

6.2 Avaliação de IOCs recolhidos

Durante o processo de desenvolvimento da Zwerg foram efectuados testes para validar o real valor dos IOCs recolhidos software no contexto do SOC do MAI. Foram também realizados testes manuais para aferir: a classificação atribuída a cada um dos indicadores; a classificação dos metadados criados que permitem associar cada um dos indicadores a um determinado tipo de ataque informático, através das etiquetas. Estes testes vieram confirmar que a maioria dos indicadores está correctamente classificada e não está presente nas listas actualmente usadas pelo MAI e que é grande o valor acrescentado pela Zwerg a todo o processo, já existente de OSINT e recolha de indicadores de compromisso.

Capítulo 7 **Conclusões e trabalho futuro**

Neste capítulo discutimos as conclusões do processo de elaboração deste projecto, desde a fase inicial de estudo sobre o estado da arte até à conclusão do projecto. Nesta discussão serão abordadas as principais problemáticas associadas ao tema do projecto, os problemas que a Zwerg veio colmatar e o trabalho futuro que irá ser implementado neste projecto.

Na primeira etapa deste projecto foi possível perceber a realidade actual na recolha de informação OSINT, com especial foco na recolha de IOCs actuais e válidos a partir de notícias online, ambos disponibilizados a partir de *feeds* XML. Durante este período tornou-se claro a complexidade e a especificidade dos processos recolha e processamento de dados, para obtenção de metainformação, de forma automatizada, pelo que a componente de pesquisa em relação às diferentes temáticas abordadas, malware, phishing, foi feita em paralelo ao estudo realizado sobre o actual estado da arte. Ao longo do processo de desenvolvimento deste projecto foram elaboradas listas de palavras chave afectas diferentes tipos de incidentes de cibersegurança para classificação dos indicadores à posteriori. Podemos concluir que o processo de classificação dos indicadores tornou-se útil para descrever os indicadores recolhidos e armazenados no MISP. Um caso prático são as campanhas de phishing, cujo tema está normalmente descrito nos conteúdos online através de palavras chave como: domain, ip, scheme, bank credentials, entre outros termos mais técnicos e específicos. Consideramos que a elaboração desta lista numa etapa inicial foi benéfica pois o tempo remanescente de desenvolvimento deste projecto foi útil para enriquecimento destas bases de dados de palavras chave.

No processo de desenvolvimento da Zwerg fomos confrontados com diversos problemas relacionados com o tema de recolha de metainformação, tais como: as aplicações externas para recolher metadados sobre os IOCs; planos de utilização e número máximo de pedidos a efectuar diariamente; tratamento de erros como consequência de utilização de planos gratuitos; identificação precisa do tipo ataque ou incidente associado a cada um dos IOCs recolhidos. O processo de resolução deste tipo de problemas foi sempre resolvido de acordo com os parâmetros utilizados no COSI.

Na resolução do problema afecto à recolha de metadados sobre cada um dos indicadores optamos por usar as APIs do Apility e VirusTotal com keys gratuitas, o que levou a que fosse necessário desenvolver mecanismos extra para os casos em que o número de pedidos à API gratuita é superior ao limite máximo diário.

O tratamento de utilização excessiva através da interpretação de resposta foi o mecanismo utilizado para evitar erros durante a execução do software. Para evitar que o número de pedidos diários, à API do Apility, fosse excedido apenas são analisados os IPs que nunca foram analisados. No entanto, no cenário de existir um dia em que sejam recolhidos mais IPs do que aqueles que são possíveis de analisar diariamente através do plano gratuito da API do Apility, são efectuados tratamentos de erros através da interpretação do código de erro presente na resposta ao pedido

de análise. Esta interpretação é tratada através de mecanismos de excepção para evitar que o software deixe de executar os restantes procedimentos, tais como classificação dos IOCs e integração dos mesmos na plataforma MISP. Este processo demonstrou ser o mais eficiente no tratamento de possíveis erros dentro dos casos de uso possíveis.

Para solucionar o problema de classificação do tipo de ataque associado a um indicador foram definidos procedimentos de busca, na notícia onde o IOC foi encontrado, que permitem procurar por palavras chaves que estão presentes numa lista de palavras chave definida em conjunto com o COSI. O objectivo destes procedimentos de busca e classificação é associar a cada indicador 3 etiquetas. Caso estes procedimentos devolvessem mais do que 3 palavras chave para um IOC, são utilizadas as etiquetas com maior número de ocorrências. Este mecanismo, provou-se eficiente após serem realizados testes, a diferentes tipos de indicadores, que tinham como objectivo definir se as etiquetas associadas a cada um dos indicadores descreviam, dentro do possível, o âmbito e contexto do IOC.

A Zwerg distingue-se dos restantes softwares existentes no mercado por ser extremamente eficiente nos indicadores recolhidos e na criação de metadados, configurável ao nível dos temas abordados e completamente autónoma em todo o seu processo. Todos os objectivos definidos pelo COSI foram cumpridos e validados durante a fase de testes, cumprindo as metas inicialmente estabelecidas

Apesar de todos os reveses encontrados, consideramos que resolvemos de forma eficaz todos os problemas encontrados durante o desenvolvimento do projecto. Como trabalho futuro serão desenvolvidos novos módulos de avaliação dos indicadores, mantendo os parâmetros criados neste projecto.

Bibliografia

[1] G. E. Moore, “Cramming more components onto integrated circuits,” *Electron. Number 8, April 19, 1965*, vol. 38, 1965.

[2] Internet Live Stats, “Internet Live Stats - Internet Usage & Social Media Statistics,” 2018. [Online]. Available: http://www.internetlivestats.com/. [Accessed: 18-Oct-2018].

[3] P. Baumard, “Cybersecurity in France,” pp. 17–31, 2017.

[4] C. Details, “CVE Vulnerabilities,” 2018. [Online]. Available: https://www.cvedetails.com/browse-by-date.php.

[5] Accenture, “COST OF CYBER CRIME STUDY 2017 INSIGHTS ON THE SECURITY INVESTMENTS THAT MAKE A DIFFERENCE Independently conducted by Ponemon Institute LLC and jointly developed by Accenture,” 2017.

[6] D. Harley, “Cybersecurity Trends 2018 : the Cost of Our Connected World,” *Eset*, pp. 1–30, 2018.

[7] Diário da República, “1ª série – N.º – 5 de Junho de 2019 – Resolução do Conselho de Ministros n.º 92/2019,” pp. 2888–2895, 2020.

[8] Symantec Corporation, “ISTR Internet Security Threat Report,” p. 89, 2017.

[9] ENISA, “Existing taxonomies,” 2018. [Online]. Available: https://www.enisa.europa.eu/topics/csirt-cert-services/community-projects/existing-taxonomies.

[10] Wikipédia, “Robert Tappan Morris,” 2018. [Online]. Available: https://en.wikipedia.org/wiki/Robert_Tappan_Morris. [Accessed: 18-Oct-2018].

[11] “RSS,” 2018. [Online]. Available: https://pt.wikipedia.org/wiki/RSS.

[12] “Feedly.” [Online]. Available: https://feedly.com.

[13] “News Blur.”

[14] C. Corp, D. Karrenberg, and E. Lear, “Rfc 1918,” *Rfc 1918*, 1996. [Online]. Available: https://tools.ietf.org/pdf/rfc1918.pdf.

[15] MongoDB, “The most popular database for modern apps - Query operator regex,” 2019. [Online].

[16] “RFC - 1630.” [Online]. Available: https://tools.ietf.org/pdf/rfc1630.pdf.

Anexos

Anexo A Ambiente de execução

Este subcapítulo enumera todos os requisitos necessários para o ambiente de execução da Zwerg no contexto do SOC do MAI. A lista de requisitos foi elaborada à data de término da conclusão deste projecto.

1. Sistema Operativo Linux
2. Firefox versão 66.0.3
3. MISP versão 2.4.107
4. Python versão 3.7.3
5. Bibliotecas de Python
    i. requests.py versão 0.4.1
    ii. pymongo.py versão 3.7.2
    iii. pprint.py versão 3.7.3
    iv. os.py – versão 3.7.3
    v. sys.py - versão 3.7.3
    vi. feedparser.py versão 5.2.1
    vii. htmlparser.py versão 3.7.3
    viii. datetime.py versão 4.3
    ix. tqdm.py versão 4.31.1
    x. re.py versão 3.7.3
    xi. termcolor.py versão 1.1.0
    xii. colorama.py versão 0.4.1
    xiii. pymisp.py versão 2.4.106

Anexo B Lista de feeds

Lista de *feeds* utilizada para a recolha de notícias:

1. https://www.schneier.com/blog/atom.xml
2. https://www.sans.org/tip-of-the-day/rss
3. https://www.webroot.com/blog/feed
4. https://www.techworld.com/security/rss
5. http://www.zonealarm.com/blog/index.php/feed/
6. https://www.proofpoint.com/us/rss.xml
7. https://www.troyhunt.com/feed
8. https://blog.pcisecuritystandards.org/rss.xml
9. http://www.veracode.com/blog/feed/
10. https://www.tripwire.com/state-of-security/feed/
11. https://sensorstechforum.com/feed/
12. https://heimdalsecurity.com/blog/feed/
13. https://www.helpnetsecurity.com/feed
14. https://www.itgovernance.co.uk/blog/category/cyber-security/feed/
15. https://blogs.seqrite.com/feed/
16. https://www.datasunrise.com/feed
17. https://itsecuritycentral.teramind.co/feed/
18. https://www.cloudbric.com/feed/
19. https://hackingvision.com/feed
20. https://hackercombat.com/feed/
21. http://www.nationalcybersecurityinstitute.org/feed/
22. https://blog.securityinnovation.com/rss.xml
23. https://www.cloudmask.com/blog/rss.xml
24. http://www.mytopposts.com/category/cyber-security/feed
25. http://securityweekly.com/podcast/psw.xml
26. https://community.connection.com/author/stephen-nardone/feed/
27. http://bhconsulting.ie/securitywatch/feed/
28. https://taosecurity.blogspot.com/feeds/posts/default?alt=rss
29. https://www.cyberdb.co/blog/feed/
30. http://feeds.feedburner.com/blogspot/CqwP
31. http://www.compasscyber.com/blog/feed/
32. https://binaryblogger.com/feed/
33. https://adamlevin.com/feed/
34. https://graquantum.com/feed/
35. https://www.lastwatchdog.com/feed/
36. https://blog.itsecurityexpert.co.uk/feeds/posts/default?alt=atom
37. http://privacyref.com/wordpress/feed/
38. http://ykileo.blogspot.com/feeds/posts/default?alt=rss
39. https://www.cyber-security-blog.com/feeds/posts/default?alt=rss
40. https://software-security.sans.org/blog/feed/
41. https://www.darkreading.com/rss_simple.asp?f_n=644&f_ln=Attack/Breaches
42. https://www.darkreading.com/rss_simple.asp?f_n=645&f_ln=Application%20Security
43. https://www.darkreading.com/rss_simple.asp?f_n=646&f_ln=Database%20Security
44. https://www.darkreading.com/rss_simple.asp?f_n=647&f_ln=Cloud
45. https://www.darkreading.com/rss_simple.asp?f_n=648&f_ln=Endpoint
46. https://www.darkreading.com/rss_simple.asp?f_n=649&f_ln=Authentication
47. http://www.informationweek.com/whitepaper/Security?gset=yes
48. https://www.darkreading.com/rss_simple.asp?f_n=664&f_ln=Vulnerability%20Management
49. https://www.darkreading.com/rss_simple.asp?f_n=663&f_ln=Insider%20Threats
50. https://www.darkreading.com/rss_simple.asp?f_n=662&f_ln=Advanced%20Threats

51. https://www.darkreading.com/rss_simple.asp?f_n=661&f_ln=Vulnerabilities%20/%20Threats
52. https://www.darkreading.com/rss_simple.asp?f_n=660&f_ln=Security%20Monitoring
53. https://www.darkreading.com/rss_simple.asp?f_n=659&f_ln=Threat%20Intelligence
54. https://www.darkreading.com/rss_simple.asp?f_n=658&f_ln=Analytics
55. https://www.darkreading.com/rss_simple.asp?f_n=655&f_ln=Operations
56. https://www.us-cert.gov/ncas/current-activity.xml
57. https://krebsonsecurity.com/feed/
58. https://vuldb.com/?rss.updates
59. https://securingtomorrow.mcafee.com/feed/
60. https://www.darkreading.com/rss_simple.asp
61. https://threatpost.com/feed/
62. https://nakedsecurity.sophos.com/feed/
63. https://blogs.quickheal.com/feed/
64. https://www.grahamcluley.com/feed/
65. https://www.infosecurity-magazine.com/rss/news/
66. https://www.csoonline.com/index.rss
67. https://www.symantec.com/connect/rss/v1/blogs/rss.xml
68. http://securityaffairs.co/wordpress/feed
69. https://www.cio.com/category/security/index.rss
70. https://cloudblogs.microsoft.com/microsoftsecure/category/cybersecurity/feed/
71. https://www.paypal.com/stories/rest/blog/rss/us
72. https://www.theguardian.com/technology/data-computer-security/rss
73. https://www.forbes.com/security/feed/
74. http://blogs.cisco.com/category/security/feed/
75. https://www.zdnet.com/blog/security/rss.xml
76. https://www.techrepublic.com/rssfeeds/topic/security/?feedType=ssfeeds
77. https://www.nist.gov/blogs/i-think,-therefore-iam/rss.xml
78. https://blog.trendmicro.com/feed/
79. http://www.truste.com/blog/feed/
80. https://www.darkreading.com/rss_simple.asp?f_n=654&f_ln=Compliance
81. https://www.darkreading.com/rss_simple.asp?f_n=653&f_ln=Risk
82. https://www.darkreading.com/rss_simple.asp?f_n=652&f_ln=Perimter
83. https://www.darkreading.com/rss_simple.asp?f_n=651&f_ln=Mobile
84. https://www.darkreading.com/rss_simple.asp?f_n=650&f_ln=Privacy
85. https://inquest.net/blog/rss
86. http://www.carbonblack.com/feed/
87. http://feeds.feedburner.com/Unit42
88. https://securelist.com/feed/
89. https://www.volexity.com/blog/feed/
90. https://www.fireeye.com/blog/threat-research/_jcr_content.feed
91. http://feeds.feedburner.com/threatconnect-blogs?format=xml
92. http://baesystemsai.blogspot.com/feeds/posts/default?alt=rss
93. http://feeds.feedburner.com/feedburner/Talos
94. http://www.clearskysec.com/feed/
95. https://www.anomali.com/site/blog-rss
96. https://www.volexity.com/blog/feed/
97. https://blog.malwarebytes.com/feed/
98. http://blog.malwaremustdie.org/atom.xml
99. https://www.us-cert.gov/ncas/alerts.xml
100. https://blog.fox-it.com/feed/
101. https://citizenlab.ca/feed/
102. http://blog.shadowserver.org/feed/
103. https://www.riskiq.com/feed/
104. http://thembits.blogspot.com/feeds/posts/default

105. http://feeds.trendmicro.com/Anti-MalwareBlog/
106. http://techhelplist.com/index.php/spam-list?format=feed&type=rss
107. http://www.vxsecurity.sg/feed/
108. https://malwarebreakdown.com/feed/

Anexo C Lista de Palavras chave

Lista de palavras chave utilizada para classificação dos IOCs:

1. APT
2. Adware
3. Backdoor
4. Botnet
5. Brute Force
6. Buffer Overflow
7. Cross-Site
8. XSS
9. Cryptography
10. Cyberespionage
11. Cyberwarfare
12. Leak
13. Leakage
14. DoS
15. DDoS
16. Exfiltration
17. Eavesdropping
18. Encryption
19. Exploit
20. Jitter
21. Malware
22. Minning
23. Payload
24. Pharming
25. Phishing
26. Sweep
27. Rootkit
28. Scanner
29. Spam
30. SQL Injection
31. SYN Flood
32. Covert
33. Portscan
34. Scan
35. UDP Scan
36. Zero Day
37. Zombies

Anexo D Screenshots da Aplicação Web

Neste anexo são apresentadas algumas figuras para visualização da Aplicação Web desenvolvida.

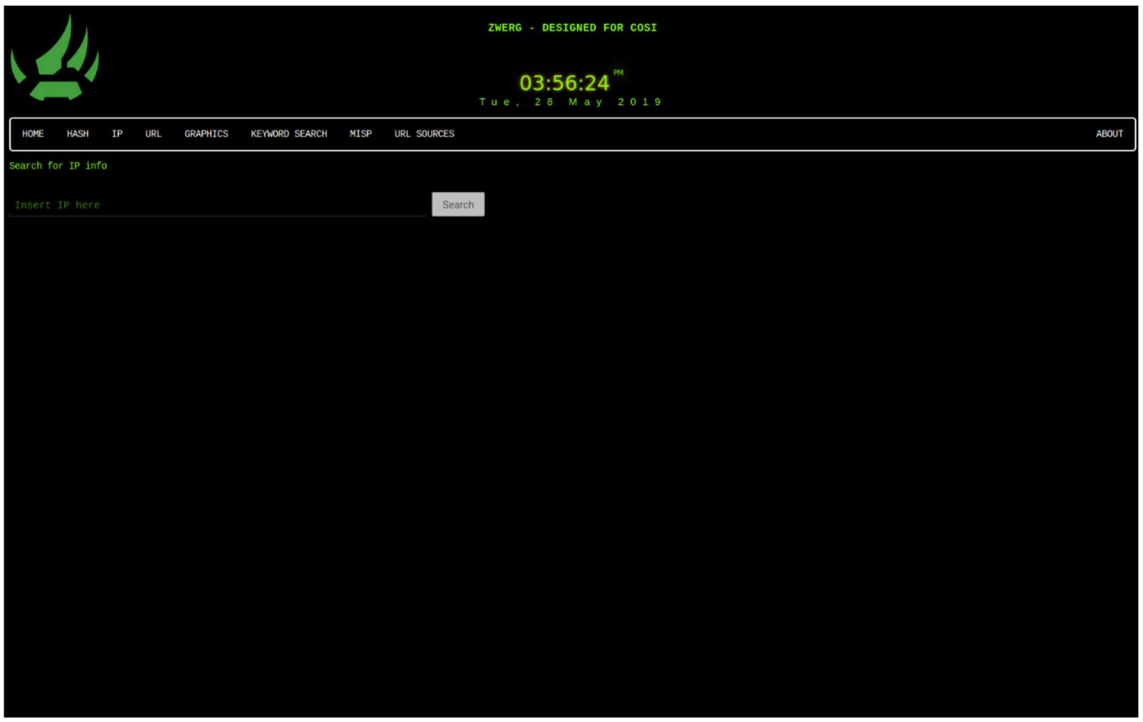

Figura D.1 - Página de pesquisa por IP.

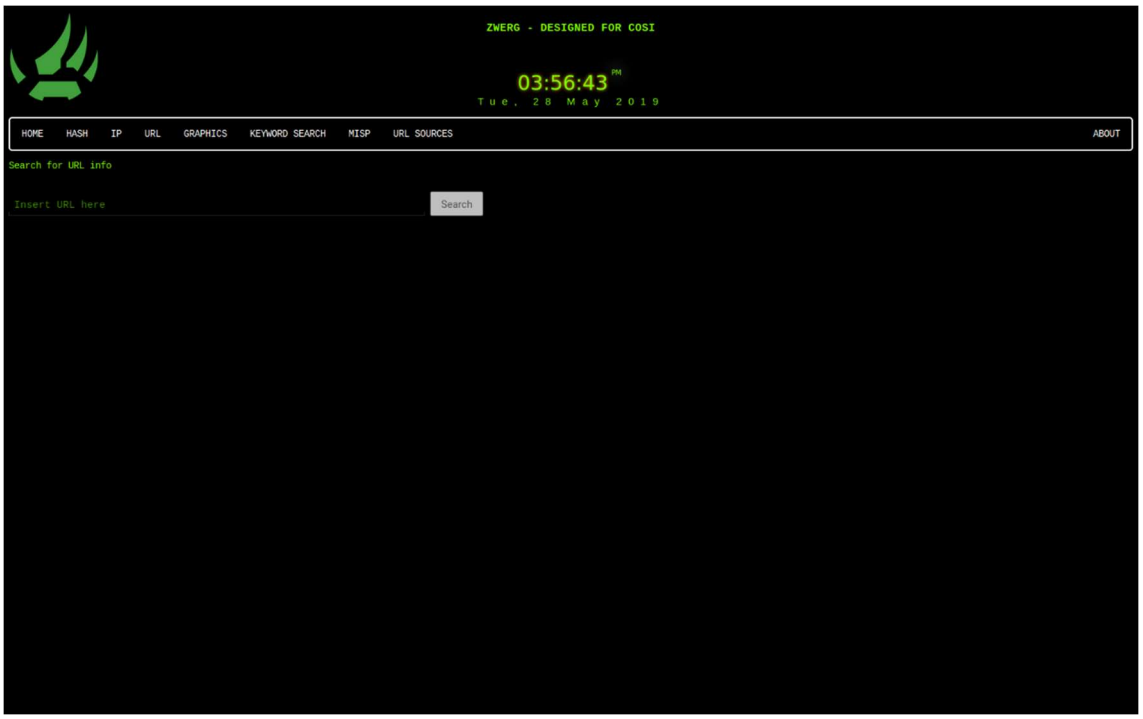

Figura D.2 - Página de pesquisa por URL ou domínio.

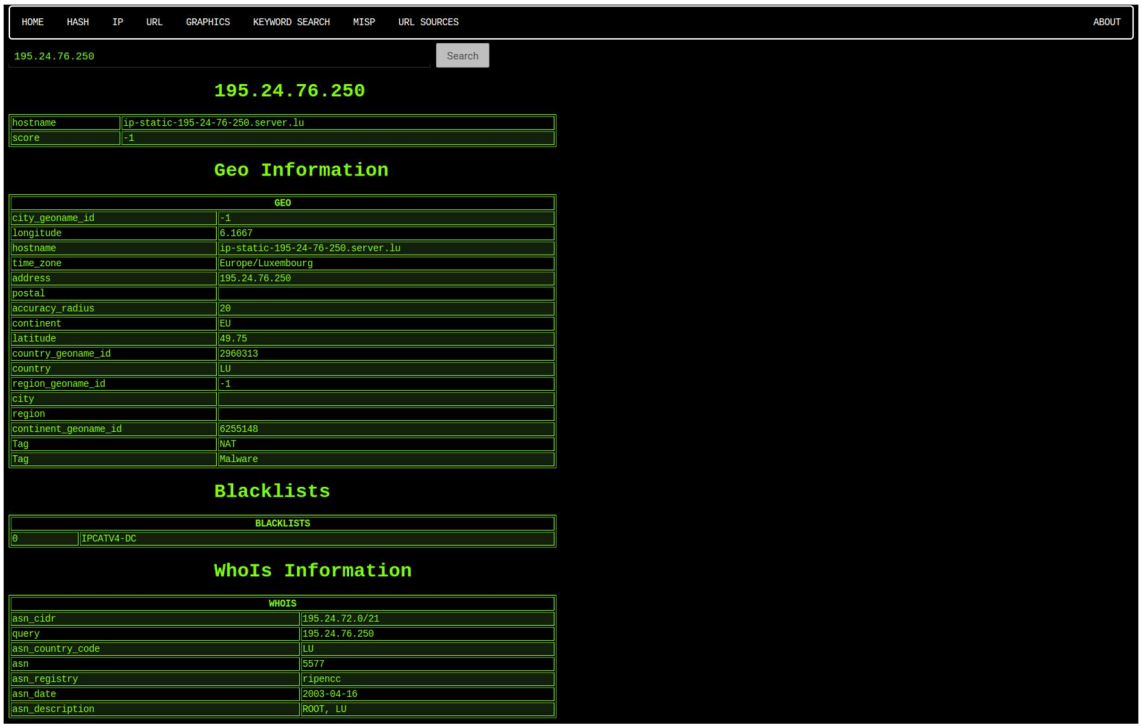

Figura D.3 - Resultados da pesquisa por IP.

Figura D.4 - Resultados da pesquisa por domínio ou URL.

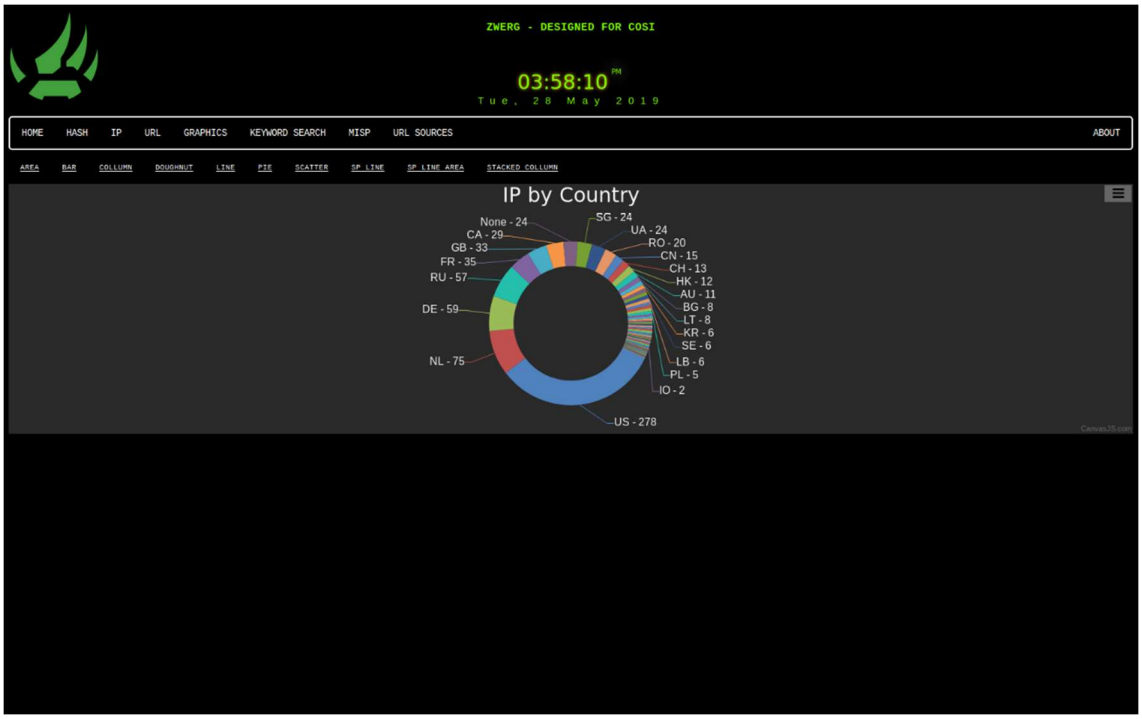

Figura D.5 - Página dos gráficos.